WORKING PAPER SERIES

FISHERIES STOCK ASSESSMENT

TITLE XII
Collaborative Research Support Program

Fisheries Stock Assessment CRSP Management Office
International Programs, College of Agriculture
The University of Maryland, College Park, Maryland 20742

In cooperation with the United States Agency for International Development (Grant No. DAN-4146-G-SS-5071-00) the Fisheries Stock Assessment CRSP involves the following participating institutions:

The University of Maryland—Center for Environmental and Estuarine Studies
The University of Rhode Island—International Center for Marine Resource Development
The University of Washington—Center for Quantitative Sciences
The University of Costa Rica—Centro de Investigación en Ciencias del Mar y Limnología
The University of the Philippines—Marine Science Institute (Diliman)—College of Fisheries (Visayas)

In collaboration with The University of Delaware; The University of Maryland—College of Business and Management; The University of Miami; and The International Center for Living Aquatic Resources Management (ICLARM).

WORKING PAPER SERIES

Working Paper No. 34

"User's Guide to CORECS - COntinuous RECruitment Simulation Model"

by

Jerald S. Ault
William W. Fox, Jr.
University of Miami

March, 1988

Fisheries Stock Assessment
Title XII
Collaborative Research Support Program

The Fisheries Stock Assessment CRSP (sponsored in part by USAID Grant No. DAN-4146-G-SS-5071-00) is intended to support collaborative research between the U.S. and developing countries' universities and research institutions on fisheries stock assessment and management strategies.

This Working Paper was produced by the University of Maryland-Center for Environmental and Estuarine Studies and the University of Costa Rica-Centro de Investigacion en Ciencias del Mar y Limnologia (CIMAR) in association with the University of Delaware and the University of Miami. Additional copies are available from the CRSP Management Office and from:

Chesapeake Biological Laboratory
Center for Environmental and Estuarine Studies
The University of Maryland
P.O. Box 38
Solomons, Maryland 20688

Fisheries Stock Assessment CRSP

USER's GUIDE To
CORECS - COntinuous RECruitment Simulation model

by

Jerald S. Ault and William W. Fox, Jr.

Division of Biology and Living Resources
Rosenstiel School of Marine and Atmospheric Science
University of Miami
4600 Rickenbacker Causeway
Miami, Florida 33149
USA

February 1988

I. Identification

Program Name: **CORECS** Version 1.0

Language: FORTRAN

Reference: Ault, J.S. 1988. *Nonlinear Numerical Simulation Models for Assessing Tropical Fisheries with Continuously Breeding Multicohort Populations*. Ph.D. Dissertation. University of Miami, Florida. 242p.

A generalized first-order age-independent numerical model that incorporates probability relations for recruitment, growth and survivorship for simulating autonomous n-cohort population tropical fishery dynamics. Coded in VAX FORTRAN V4.5-219 for the Digital mainframe, and FORTRAN 77 for the COMPAQ 286 and IBM-PC/AT microcomputers.

II. Introduction

CORECS is a widely applicable finite time-step (Δt), numerical population simulation model. The model was designed for applications to a tropical context by utilizing continuous versus discrete decision variables. It was specifically designed to produce stochastic age-independent population probability density distributions peculiar to the specific continuous spawning and recruiting life history patterns typical of tropical and subtropical marine fish stocks. Computations are performed on any portion of a whole time unit (typically 1-year) desired. Computations are summarized per Δt, and also over the whole unit. Numerical ASCII files are written to a flexible graphics routine that allows two-plane plots of any of the population processes desired. Components of the model programmed into the present version are: (i) a first-order dynamical difference equations scheme which allows any time stream division desired, (ii) continuous propagation of cohorts following various probability density functions, (iii) stochastic representations of age-specific growth and survivorships allowing age-independent length frequency distributions, (iv) age- and sex-specific maturation schedules, (v) linkage of the growth and reproductive processes, and (vi) a computation scheme that allows for seasonal mortalities and any specified partial or seasonal selectivity pattern.

III. Input Data File Setup

Data files dimensioned as 8F10.0.

Integer Variables (All values are Integer I10):

Row 1

Var. #	Variable	Definition
1	NDT ≡	Number of finite divisions of the annual rate values, thus Δt = 1/NDT.
2	MOS ≡	Number of finite Δt's post-equilibrium.
3	NTL ≡	Oldest annual age class carried in the simulation (NM = NTL * NDT ≡ number of population cohorts).
4	MFR ≡	Time-step of cohort life during which recruitment to the exploitable phase of the stock occurs.
5	MFC ≡	Time-step of cohort life for first capture by the fishing gear.
6	MRM ≡	Time-step of cohort first reproductive maturity.
7	NFOT ≡	Type of Fishing Mortality Pattern desired. 1.) Pope's (1972) mid-year approximation. 2.) Constant fishing mortality for cohorts and by Δt. 3.) Seasonal fishing mortality strategy.
8	NROT ≡	Type of Recruitment Pattern desired. 1.) Discrete (viz. GXPOPS-type). 2.) Continuous (constant uniform per Δt). 3.) Trigonometric Distribution. 4.) Beta Distribution. 5.) Density-dependent.

Row 2

Var. #	Variable	Definition
1	NOUT ≡	Type of output desired when generating transitional conditions. 0.) Condensed. 1.) Complete per Δt.
2	NAVA ≡	Type of Availability or Selectivity pattern desired. 1.) Constant by Δt. 2.) Seasonal availability.
3	NLL ≡	Number of annual steps (Years) of simulation desired.
4	NOPT ≡	Output option.
5	NDIST ≡	number of points on the growth curve distributed at a given level of age.
6	NAMP ≡	multiplication factor for standard error of growth on age relationship.

7	NPLOT ≡	plotting array number.
8	NRSCH ≡	stochastic search (microcomputer versions only).

Real Variables (All values are Real F10.0):

Row 3

Var. #	Variable	Definition
1	ULINF ≡	Ultimate length from the von Bertalanffy formulation.
2	WINF ≡	Ultimate weight from the von Bertalanffy formulation.
3	XAKV ≡	Growth coefficient (annual rate).
4	T0 ≡	Time at which the weight/length of the fish was equal to zero.
5	XNM ≡	Annual natural mortality rate.
6	FMORT ≡	Annual fishing mortality rate.
7	ALPHA ≡	α coefficient from the fecundity on weight realtionship.
8	BETA ≡	β coefficient from the fecundity on weight realtionship.

Row 4

Var. #	Variable	Definition
1	EQUIL ≡	Generate either stationary or dynamics time vectors. 0.) Stationary. 1.) Dynamic.
2	TR ≡	Total annual recruitment in numbers of fish.
3	PER ≡	Input value for trigonometrically distributed periodic recruitment. 12) XDT - Bimodal annually with 12 time-steps per year. 24) XDT*12 - Unimodal annually with 12 time-steps per year.
4	PARM1 ≡	α_1 Parameter for the Beta Distribution.
5	PARM2 ≡	α_2 Parameter for the Beta Distribution.
6	XNMF ≡	Number of time-steps of seasonal fishing activity.
7	DTL ≡	Length increment size for grouping z elements in the nonlinear search routine.
8	ALSLB ≡	α coefficient from the stochastic power function for standard error of length on age.

Row 5

Var. #	Variable	Definition
1	BLSB ≡	β coefficient from the stochastic power function for standard error of length on age.

Row 6

Var. # Variable	Definition
FP(1) - FP(XDT) ≡	Seasonal fishing mortality switch (1=On, 0=Off).

Row 7

Var. # Variable	Definition
AM(1) - AM(XDT) ≡	Seasonal fishing mortality switch (1=On, 0=Off).

Variables for the Search Routine:

Variable	Definition
XLMAX ≡	Maximum length (upper bound) included for stochastic search.
XLMIN ≡	minimum length (lower bound) included for stochastic search.
DTL ≡	ΔLength increment for grouping the z length intervals in the search.
RANGEL ≡	total length search range.
TOTLSI ≡	total number of length increments to be searched.
XMINLEN ≡	minimum length of present search increment.
XMAXLEN ≡	maximum length of present search increment.
XLD(j,k) ≡	length associated with the jth age group and the kth probability point on the jth length distribution.
POPLEN(z) ≡	total number of individuals found within the aggregate length group z.
XLENPT(z) ≡	midpoint length of the aggregate grouping z.
TL ≡	oldest jth age increment in the stock.
DIST ≡	number of k probability points on the jth length distribution for each age-specific deterministic length.

Parameterization of CORECS

A set of life history parameters for each of several stock genera were compiled from the relevant literature to generate simulated tropical fish stock populations. They genera parameterized were: (1) grouperoids (Serranidae ≡ gag grouper), (2) scombroids (Scombridae ≡ yellowfin tuna), and (3) engrauloids (Engraulidae ≡ northern anchovy). They constituted a representative spectrum of the life histories common to tropical and subtropical fishes. The genera all exhibit quasi-continuous or protracted spawning/recruitment. Extensive information was available for development of the parameters and continuous functions required by the CORECS model. Discrete relations were fit to continuous functions to conform to the framework of CORECS.

Nonlinear Estimation of CORECS Model Parameters

The CORECS model required information based on continuous functional relationships. Generally all of the life history relationships embedded in the simulation model CORECS were nonlinear, i.e. in the sense that they involved some parameters θ_1 and θ_2 in a nonlinear way. To accommodate the model continuous functions were fit to discrete data fields by nonlinear least squares regression. The method employed here for fitting nonlinear functions of several variables involved Marquardt's (1963) compromise between the linearization method (i.e. Taylor series approximation) and the steepest descent method. The idea of Marquardt's method is explained briefly in Draper and Smith (1981). The computer program NLSQ coded in FORTRAN with subroutines MARQ, MINV, DATIN, and FUNCT was used for the purpose of fitting nonlinear functions of several variables. MARQ is a general nonlinear least squares subroutine which uses Marquardt's algorithmic modification of Newton's method, which rather than compute explicit first derivatives, approximates the first derivative by evaluating the change in the value of the function for a small, but finite, increment in the argument. The subroutine calls MINV, which computes the inverse of a square matrix, and the problem specific subroutine

FUNCT, which contains the coding for the nonlinear function that is being fit. Nonlinear estimation requires initial parameter estimates, and the better these initial estimates are, the faster will be the convergence to the fitted values. If the initial estimates are poor, convergence to the wrong final values can occur.

Grouperoids (grouper-like fishes)

Grouperoids form ecologically and economically important components of reef fish communities in tropical marine waters. Reef fishes refer to an extremely diverse category of co-occurring demersal and semi-pelagic fishes that are often exploited within multispecies complexes. Grouperoid model parameterization utilized and expanded upon the data and parameter sets found in Manooch and Haimovici (1978), Bannerot (1984), and Collins et al. (1987) for the gag grouper. The gag is a demersal marine serranid found in a variety of subtropical to tropical habitats in the western Atlantic and the Gulf of Mexico (Manooch & Haimovici, 1978). Grouperoids are top-level carnivores. Grouperoids are migratory however they tend to be habitat specific and are generally concentrated over very irregular substrates but their center of abundance is reef environment (Manooch & Haimovici, 1978). Specific populations behave as unit stocks (Bannerot, 1984). Grouperoids have a relatively high survivorship with comparatively slow growth ($M/K = 1.64$) and ultimately reach a large size. Parameters for the grouperoid stochastic growth model for were generated by fitting equation (24) of Ault and Fox (1988) to back-calculated data for total lengths of gag grouper aged by otoliths (Manooch and Haimovici, 1978). Manooch and Haimovici (op cit.) suggested there were about 13 annual classes in the exploitable phase of the life span of the gag, although Collins et al. (1987) recently suggested there may be 22 annual age groups. Gags are protogynous hermaphrodites (McErlean & Smith, 1964; Collins et al., 1987), i.e. individuals mature as females but later transform to function as males. Table 1 lists the model parameters used to represent grouperoids in the simulations.

Scombroids (tuna-like fishes)

Scombroids (Scombridae) support extensive commercial fisheries in tropical and subtropical seas worldwide. Scombroids are highly esteemed for their flesh and are thus pursued by many nation's fishing fleets making management of these species a complex task (Joseph, 1970). Parameterization of the Scombroid model focused and expanded upon the data and parameter sets found in Schaefer (1967) and Bayliff (1980) for tropical eastern Pacific yellowfin tuna. Scombroids are pelagic schooling fishes considered apex predators which inhabit the open ocean to nearshore pelagic waters. However, they tend to concentrate around frontal and upwelling zones where the greatest abundance of their food supply can be found. Scombroids are highly migratory and thus the actual definition of a unit stock may be ambiguous. Scombroids have much lower survivorship than grouperoids, and are considered fast growers ($M/K = 1.33$) which reach a relatively large size. Scombroids modeled here had 10 annual classes in the fishable life span (Schaefer, 1967). Table 2 lists the model parameters used to represent Scombroids.

Engrauloids (anchovy/sardine-like fishes)

Engrauloids are a schooling pelagic species known to concentrate in coastal frontal and upwelling regions along the continental slope, areas reknown for supporting relatively large biomasses. Generally, engrauloids are pursued by purse seines and they often support major commercial reduction fisheries where the catch is processed to provide fish meal and oil for other ancillary industries. Catches of engrauloids are important to global economics. Engrauloids are primary predators of the plankton, feeding extensively off dinoflagellate and micro-zooplankton blooms. The parameterization of the Engrauloid model focused and expanded upon the data and parameter sets of Tillman & Stadelman (1976), Ault and Broadhead (1980), and Hunter and Macewicz (1985) for the subtropical northern anchovy. Engrauloid stock structures are usually well-defined. Problems for fishery management include, but are not exclusive to, the fact that stock sizes fluctuate widely due to random environmental factors, fishing activities, and competition among the species themselves during their early life history. Recruitment varies due to biotic and gamma processes (Lasker & Smith, 1977; Lasker, 1981). Engrauloids have a very low survivorship and can be considered intermediate growers (M/K = 3.55) which reach a relatively small size. Seven annual classes are in the fishable life span (Sunada 1977, 1979; Ault & Broadhead, 1980). The stochastic growth model was generated by fitting equation (24) of Ault and Fox (1988) to the series of data for standard lengths of anchovies aged by otoliths by Sunada (1977, 1979). Table 3 lists the model parameters used to represent engrauloids.

IV. Output Data Files:

Unit	File Name	Contents
1	'Species'	Input parameters as per user's guide
2	FEC	SLT(J), PD(J), NPLOT
3	FRAC	SLT(J), YN(J), K
4	LENDIST	XLENPT(J), POPLEN(J), K
5	REPRO	J,P(I,J),SLT(J),WT(J),BIOM(J),YW(I,J),YN(I,J),F(I,J)
		PJ, BMJ, YWJ, YNJ
		AL(K), ALC(K), AVWT(K)
		EYPR, SSB
		J,FEC(J),BIT(J),BIRTHS(J),CX(J)
		FECUN,SMPFEC,ACBT,CXT
6	→Monitor Screen	
7	LENGTH	XX, XLD(J,KK), K
8	PROBDEN	XLD(J,KK), PD(J,KK), K
9	AVLENGTH	XNY, ALC(K), NPLOT
10	YLDPR	XNY, YWJ, NPLOT
		XDLENPT(J), DPOP(J), K
11	LENFREQ	XDLENPT(J), DYPD(J), KKK
		TDLC, TYD, DALC
		TPD, TYD, TSY
12	STYLDN	XLENPT(J), SYLDN(J), KKKK
		TSLC, TSY, SALC

V. Acknowledgements

This work was partially supported by a subcontract from the University of Maryland under U.S. Agency for International Development Contract No. DAN-4146-G-SS-5071-00 for the Fisheries Stock Assessment Collaborative Research Support Program.

VI. Literature Cited

Ault, J.S. 1988. *Nonlinear Numerical Simulation Models for Assessing Tropical Fisheries with Continuously Breeding Multicohort Populations*. Ph.D. Dissertation. Univ. of Miami, Florida. 242p.

Ault, J.S. and G.C. Broadhead. 1980. Status of the Baja California anchovy fishery. Living Marine Resources Rept. San Diego, Calif. 31p.

Ault, J.S. and W.W. Fox, Jr. 1988. The tropical fishery system model CORECS - COntinuous RECruitment Simulation. U.S. Agency for International Development, Fish. Stock. Assess. CRSP Working Paper., 37p.

Bannerot, S.P. 1984. *The Dynamics of Exploited Groupers (Serranidae): An Investigation of the Protogynous Hermaphroditic Reproductive Strategy.* Ph.D. Dissertation, University of Miami, Florida. 394p.

Bayliff, W. H. 1980. Synopses of biological data on eight species of scrombrids. Inter-Amer. Trop. Tuna Comm. Spec. Rpt. No. 2. 530 p.

Collins, M.R.; Waltz, C.W.; Roumillat, W.A. and D.L. Stubbs. 1987. Contribution to the life history and reproductive biology of the gag, *Mycteroperca microlepis* (Serranidae), in the South Atlantic bight. Fish. Bull. 85(3):648-653.

Draper, N. R. and H. Smith. 1981. *Applied Regression Analysis.* John Wiley & Sons. New York. 709p.

Hunter, J. R. and B. J. Macewicz. 1985. Measurement of spawning frequency in multiple spawning fishes. In R. Lasker (ed.), An egg production method for estimating spawning biomass of pelagic fish: Application to the northern anchovy, *Engraulis mordax*, pp. 79-94. U.S. Dep. Commer., NOAA Tech. Rep. NMFS 36.

Joseph, J. 1970. Management of tropical tunas in the eastern Pacific Ocean. Trans. Amer. Fish. Soc. 99(3):629-648.

Knudsen, P. K. 1977. Spawning of yellowfin tuna and the discrimination of subpopulations. IATTC Bull. 17(2).

Lasker, R., editor. 1981. *Marine Fish Larvae.* University of Washington Press. Seattle, Washington. 131p.

Lasker R. and P. E. Smith. 1977. Estimation of the effects of environmental variations on the eggs and larvae of the northern anchovy. Calif. Coop. Oceanic Fish. Invest. Rep. 19:128-137.

LeGuen, J.C. and G.T. Sakagawa. 1973. Apparent growth of yellowfin tuna from the eastern Atlantic Ocean. Fish. Bull. 71:175-187.

Manooch, C.S. and M. Haimovici. 1978. Age and growth of the gag, *Mycteroperca microlepis*, and size-age composition of the recreational catch off the southeastern United States. Trans. Am. Fish. Soc. 107(2):234-240.

Marquardt, D. W. 1963. An algorithm for least squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math. 11:431-441.

McErlean, A.J. and C.L. Smith. 1964. The age of sexual succession in the protogynous hermaphrodite, *Mycteroperca microlepis.* Trans. Am. Fish. Soc. 93(3):301-302.

Schaefer, M.B. 1967. Fishery dynamics and present status of the yellowfin tuna population of the eastern Pacific Ocean. IATTC Bull. 12(3):89-133.

Sunada, J.S. 1977. The northern anchovy fishery for the 1975-76 season. Calif. Fish Game Mar. Res. Tech. Rept. 39:1-27.

Sunada, J.S. 1979. The anchovy reduction fishery for the 1977-78 season. Calif. Fish Game Mar. Res. Tech. Rept. 43:1-24.

Tillman, M.F. and D. Stadelman. 1976. Development and example application of a simulation model of the northern anchovy fishery. Fish. Bull. 74(1):118-130.

Table 1: Life history parameters developed for the model of grouperoid stock dynamics. Data taken primarily from: (1) McErlean & Smith (1964), (2) Manooch & Haimovici (1978), (3) Bannerot (1984), (4) Collins et al. (1987), and (--) Ault (1988).

Description	Parameter	Symbol	Value	Source
Oldest annual age in the Catch	t_λ	NTL	13.0 Years 22.0	2 4
Recruitment to Exploited Phase	t_p	MFR	13.0 Months	--
Minimum age of Reproductive Maturity	t_m	MRM	49.0 Months 28%-III,51%-IV	1 4
Size/age of First Capture	t_p'	MFC	13.0 Months	--
Ultimate Length	L_∞	ULINF	1290.0(1122.6) mm	2(4)
Ultimate Weight	W_∞	WINF	25.032719(16.30) kg	2(4)
Time at which weight/ length of fish equals zero	t_0	T0	-1.127(4.0514E-3) Years	2(4)
Growth coefficient	K	XAKV	0.122(0.18343) Year^{-1}	2(4)
Natural Mortality	M	XNM	0.2 Year^{-1}	2
Fishing Mortality	F	FMORT	Variable	--
α-coefficient from fecundity on weight relation	α	ALPHA	10899.5251	--
β-coefficient from fecundity on weight relation	β	BETA	2.58077181	--
A-coefficient from weight on length relation	a	A	1.2E-08(3.971E-8)	2(4)
B-coefficient from weight on length relation	b	B	2.996(2.8238)	2(4)
α-coefficient from $\overline{L}$ stnd error on age relationship	α_{SI}	ALSLB	0.10398	--
β-coefficient from $\overline{L}$ stnd error on age relationship	β_{SI}	BSLB	0.75865	--

L_c = 304.90623
L_λ = 1059.81036

Table 1 (Continued)

Maturity Fractions ($\Theta_f \equiv$ fraction female, $\Theta_m \equiv$ fraction male).

Age	1	2	3	4	5	6	7	8	9	10	11	12	13
Θ_f	1.0	1.0	1.0	1.0	.91	.85	.70	.50	.40	.25	.10	.05	0.0
Θ_m	0.0	0.0	0.0	0.0	.09	.15	.30	.50	.60	.75	.90	.95	1.0
Source: Bannerot (1984)													
Θ_f	1.0	1.0	1.0	1.0	1.0	1.0	.98	.95	.79	.60	.44	.27	.11
Θ_m	0.0	0.0	0.0	0.0	0.0	0.0	.02	.05	.21	.40	.56	.73	.89
Source: Collins et al. (1987)													

Table 2: Life history parameters developed for the model of scombroid stock dynamics. Data taken primarily from: (1) Schaefer (1967), (2) LeGuen & Sakagawa (1973), (3) Knudsen (1977), (4) Bayliff (1980), and (--) Ault (1988).

Description	Parameter	Symbol	Value	Source
Oldest annual age in the Catch	t_λ	NTL	10.0 Years	1
Recruitment to Exploited Phase	t_ρ	MFR	11.0 Months	1
Minimum age of Reproductive Maturity	t_m	MRM	13.0 Months	3
Size/age of First Capture	t_ρ'	MFC	11.0 Months	1
Ultimate Length	L_∞	ULINF	1690.0 mm	1
Ultimate Weight	W_∞	WINF	99.1272728	1
Time at which weight/ length of fish equals zero	t_0	T0	0.83333333 Years	1
Growth coefficient	K	XAKV	0.60(0.42) Year^{-1}	1(2)
Natural Mortality	M	XNM	0.80 Year^{-1}	1
Fishing Mortality	F	FMORT	Variable	--
α-coefficient from fecundity on weight relation	α	ALPHA	13226.0815	--
β-coefficient from fecundity on weight relation	β	BETA	1.44288696	--
A-coefficient from weight on length relation	a	A	3.894E-08	4
B-coefficient from weight on length relation	b	B	3.020	4
α-coefficient from $\overline{L}$ stnd error on age relationship	α_{SI}	ALSLB	0.10398	--
β-coefficient from $\overline{L}$ stnd error on age relationship	β_{SI}	BSLB	0.75865	--

L_c = 82.42223078(67.3480052)
L_λ = 1683.09336(1670.71842)

Table 3: Life history parameters developed for the model of engrauloid stock dynamics. Data taken primarily from: (1) Ault & Broadhead (1980), (2) Tillman & Stadelman (1976), (3) Sunada (1977, 1979), (4) Hunter & Macewicz (1985), and (--) Ault (1988).

Description	Parameter	Symbol	Value	Source
Oldest annual age in the Catch	t_λ	NTL	7.0 Years	3
Recruitment to Exploited Phase	t_ρ	MFR	7.0 Months	1
Minimum age of Reproductive Maturity	t_m	MRM	6.0 Months	4
Size/age of First Capture	t_ρ'	MFC	7.0 Months	1
Ultimate Length	L_∞	ULINF	159.1 mm	2
Ultimate Weight	W_∞	WINF	4.6826E-2 kg.	--
Time at which weight/ length of fish equals zero	t_0	T0	-2.08 Years	2
Growth coefficient	K	XAKV	0.32 $Year^{-1}$	2
Natural Mortality	M	XNM	0.80 $Year^{-1}$	1
Fishing Mortality	F	FMORT	Variable	--
α-coefficient from fecundity on weight relation	α	ALPHA	111.75	2 & --
β-coefficient from fecundity on weight relation	β	BETA	1.47632	2 & --
A-coefficient from weight on length relation	a	A	1.21E-5	1
B-coefficient from weight on length relation	b	B	2.9733	1
α-coefficient from $\overline{L}$ stnd error on age relationship	α_{SI}	ALSLB	0.026947	3 & --
β-coefficient from $\overline{L}$ stnd error on age relationship	β_{SI}	BSLB	0.7.593E-1	3 & --

$L_c = 91.2522463$
$L_\lambda = 150.394692$

```
$DEBUG
      PROGRAM COREX
C*********************************************************************
C                                                                    *
C     AULT, JERALD S. 1987.                                          *
C     PROGRAM CORECS CODED IN FORTRAN 77 FOR THE COMPAQ 286.         *
C     PROGRAM TO SIMULATE A QUASI-CONTINUOUS RECRUITING LIFE         *
C     HISTORY TYPICAL OF TROPICAL MARINE FISH STOCKS.                *
C                                                                    *
C*********************************************************************
      INTEGER ZZ,X,SMOS
      DIMENSION XM(156),A(156),F(156),P(156,4),R(170),
     *  SLT(156),WT(156),BIOM(156),YW(156),
     *  RN(156),YN(156),FM(12),SXR(12),XMU(12),
     *  ALC(156),Z(156),XE(156),RA(30),RC(30),AVWT(156),
     *  FEC(156),Z1(156),
     *  FMULT(12),
     *  AP(156),AYW(156),AYN(156),FP(12),AM(12),
     *  FRQ(31),SE(156),XLD(135,31),
     *  XLENPT(100),
     *  PDY(135,31),SYLND(100)
      OPEN(UNIT=12,FILE='STYLDN.DAT',STATUS='NEW')
C      OPEN(UNIT=11,FILE='LENFREQ.DAT',STATUS='NEW')
1      OPEN(UNIT=10,FILE='YLDPR.DAT',STATUS='NEW')
C      OPEN(UNIT=9,FILE='AVLENGTH.DAT',STATUS='NEW')
C      OPEN(UNIT=8,FILE='PROBDEN.DAT',STATUS='NEW')
C      OPEN(UNIT=7,FILE='LENGTH.DAT',STATUS='NEW')
      OPEN(UNIT=5,FILE='REPRO.DAT',STATUS='NEW')
      OPEN(UNIT=4,FILE='LENDIST.DAT',STATUS='NEW')
C      OPEN(UNIT=3,FILE='FRAC.DAT',STATUS='NEW')
C      OPEN(UNIT=2,FILE='FEC.DAT',STATUS='NEW')
      OPEN(UNIT=1,FILE=' ',STATUS='OLD')
      WRITE(*,30)
   30 FORMAT(7X,'INITIALIZING CONTINUOUS TIME DATA ARRAYS',//)
      READ(1,31)NDT,MOS,NTL,MFR,MFC,MRM,NFOT,NROT,
     *          NOUT,NAVA,NLL,NOPT,NDIST,NAMP,NPLOT,NSRCH
      READ(1,29)ULINF,WINF,XAKV,T0,XNM,FMORT,ALPHA,BETA,
     *          EQUIL,TR,PER,PARM1,PARM2,XNMF,DTL,ALSLB
      READ(1,33)BSLB,A4,B4
      READ(1,29)FP(1),FP(2),FP(3),FP(4),FP(5),FP(6),FP(7),FP(8),
     *          FP(8),FP(9),FP(10),FP(11),FP(12)
      READ(1,29)AM(1),AM(2),AM(3),AM(4),AM(5),AM(6),AM(7),AM(8),
     *          AM(9),AM(10),AM(11),AM(12)
      READ(1,29)(FRQ(I),I=1,NDIST)
   31 FORMAT(8I10)
   33 FORMAT(3F15.0)
   29 FORMAT(8F10.0)
C*********************************************************************
C     INITIALIZE CONTINUOUS SIMULATION ARRAYS                        *
C*********************************************************************
      XDT=NDT
      DT=1/XDT
      NM=NTL*NDT
      NMC=NM+1
      NMT=NM+MOS
```

```
      XMU(1)=0.0
      FMULT(1)=1.0
      XAKV=XAKV/XDT
      T0=T0*XDT
      J=1
      DO 199 I=1,NM
      XE(J)=1.-EXP(-XAKV*((J-0.5)-T0))
      IF(XE(J))191,191,192
  192 SLT(J)=ULINF*XE(J)
      WT(J)=A4*(SLT(J)**B4)
      IF(J.LT.MRM) GOTO 193
      FEC(J)=ALPHA*(WT(J)**BETA)
      GOTO 198
  191 SLT(J)=0.0
      WT(J)=0.0
  193 FEC(J)=0.0
  198 J=J+1
  199 CONTINUE
      DO 55 I=1,NM
      SE(I)=ALSLB*((I)**BSLB)
   55 CONTINUE
C*********************************************************************
C     NATURAL MORTALITY ARRAY                                        *
C*********************************************************************
      DO 32 J=1,(MFR-1)
      XM(J)=0.0
   32 CONTINUE
C      XM(45)=(XNM/XDT)*0.36
C
      DO 35 J=MFR,NMT
      XM(J)=XNM
   35 CONTINUE
      GOTO 92
C*********************************************************************
C     AVAILABILITY ARRAY                                             *
C*********************************************************************
C     SELECTIVITY PATTERNS                                           *
C     VARIABLE NAME = NAVA                                           *
C                                                                    *
C     CONSTANT FOR ALL INTERVALS = 1                                 *
C     SEASONALLY AVAILABLE = 2                                       *
C*********************************************************************
   92 IF(NAVA.EQ.1) GOTO 41
      IF(NAVA.EQ.2) GOTO 91
   41 DO 42 J=1,(MFC-1)
      A(J)=0.0
   42 CONTINUE
      DO 45 J=MFC,NM
      A(J)=1.0
   45 CONTINUE
      GOTO 200
C*********************************************************************
C     SEASONAL AVAILABILITY                                          *
C*********************************************************************
   91 DO 107 I=1,NLL
```

```
      DO 106 M=1,12
      K=(I-1)*12+M
      IF(AM(M).EQ.0) GOTO 95
      IF(AM(M).EQ.1) GOTO 97
   95 X=K
      J=1
      DO 96 JJ=X,1,-1
      IF(J.GT.NMT) GOTO 96
      A(J)=0.0
      J=J+1
   96 CONTINUE
      GOTO 106
   97 X=K
      J=1
      DO 98 JJ=X,1,-1
      IF(J.GT.NMT) GOTO 98
      A(J)=1.0
      J=J+1
   98 CONTINUE
  106 CONTINUE
  107 CONTINUE
      GOTO 200
C***********************************************************************
C     BIRTH RATE FUNCTION ARRAYS                                       *
C***********************************************************************
C     CALCULATION OF AGE(SIZE) SPECIFIC BIRTH RATES                    *
C     (NET MATERNITY LxMx) BY MANIPULATION OF THE                      *
C     FECUNDITY ON WEIGHT RELATIONSHIP                                 *
C***********************************************************************
  200 RHO=0.8
C     XLAMBDA=0.8
C     FECUN=0.0
C     SMPFEC=0.0
C     DO 201 JJ=1,MRM
C     FEC(JJ)=0.0
C 201 CONTINUE
C     DO 206 JJ=MRM,NM
C     FEC(JJ)=ALPHA*(WT(JJ)**BETA)
C     IF(RHO.NE.XLAMBDA) GOTO 202
C     IF(RHO.EQ.XLAMBDA) GOTO 203
C 202 BIT(JJ)=FEC(JJ)*(((XLAMBDA**(JJ+1-MRM))-(RHO**
C    *          (JJ+1-MRM)))/(RHO-XLAMBDA))
C     GOTO 205
C 203 BIT(JJ)=FEC(JJ)*(JJ+1-MRM)*(RHO**(JJ-MRM))
C     GOTO 205
C 205 BIRTHS(JJ)=BIT(JJ)*((EXP(-XM(JJ)))**(JJ-1))
C     FECUN=FECUN+FEC(JJ)
C 206 CONTINUE
C     DO 207 JJ=1,NM
C     PBIRTH(JJ)=FEC(JJ)/FECUN
C     PBIRTH(JJ)=PBIRTH(JJ)*100.
C     SMPFEC=SMPFEC+PBIRTH(JJ)
C 207 CONTINUE
C***********************************************************************
C     FISHING MORTALITY ARRAY                                          *
```

```
C**********************************************************************
C      FISHING MORTALITY PATTERNS                                     *
C      VARIABLE NAME = NFOT                                           *
C      POPE'S (1971) METHOD = 1                                       *
C      CONSTANT FOR ALL INTERVALS = 2                                 *
C      SEASONAL FISHING STRATEGY = 3                                  *
C**********************************************************************
      IF(NFOT.EQ.1) GOTO 21
      IF(NFOT.EQ.2) GOTO 60
      IF(NFOT.EQ.3) GOTO 130
C*****POPE'S MID-YEAR APPROXIMATION
   21 DO 20 I=1,NMT
      DO 19 L=1,14
      L1=0
      L1=(L-1)*12
      IF(I.EQ.L1) GOTO 18
      IF(I.NE.L1) GOTO 19
   18 DO 17 J=1,NMT
      F(J)=FMORT
   17 CONTINUE
      GOTO 20
   19 CONTINUE
      DO 15 J=1,NMT
      F(J)=0.
   15 CONTINUE
   20 CONTINUE
      GOTO 84
C*****F CONSTANT EVERY INTERVAL
   60 DO 61 J=1,(MFC-1)
      F(J)=0.
   61 CONTINUE
C       F(45)=(FMORT)*0.36
      DO 64 J=MFC,NMT
      F(J)=FMORT
   64 CONTINUE
      GOTO 84
C*****SEASONAL FISHING MORTALITY STRATEGY
  130 DO 149 I=1,NLL
      DO 148 M=1,12
      K=(I-1)*12+M
      IF(FP(M).EQ.0) GOTO 131
      IF(FP(M).EQ.1) GOTO 135
  131 X=K
      J=1
      DO 141 JJ=X,1,-1
      IF(J.GT.NMT) GOTO 141
      F(J)=0.
      J=J+1
  141 CONTINUE
      GOTO 148
  135 X=K
      J=1
      DO 142 JJ=X,1,-1
      IF(J.GT.NMT) GOTO 142
      F(J)=FMORT/XNMF
```

```
      J=J+1
  142 CONTINUE
  148 CONTINUE
  149 CONTINUE
      GOTO 84
C*********************************************************************
C     RECRUITMENT ARRAYS                                              *
C*********************************************************************
C     RECRUITMENT DISTRIBUTION FUNCTIONS                              *
C     VARIABLE NAME = NROT                                            *
C                                                                     *
C     DISCRETE = 1                                                    *
C     CONTINUOUS UNIFORM = 2                                          *
C     TRIGONOMETRIC DISTRIBUTION = 3                                  *
C     BETA DISTRIBUTION = 4                                           *
C                                                                     *
C     TR = TOTAL ANNUAL MAGNITUDE OF RECRUITMENT                      *
C*********************************************************************
   84 IF(NROT.EQ.1) GOTO 86
      IF(NROT.EQ.2) GOTO 87
      IF(NROT.EQ.3) GOTO 88
      IF(NROT.EQ.4) GOTO 90
C*****DISCRETE ANNUAL RECRUITMENT
   86 DO 71 I=1,NTL
      DO 70 J=1,NDT
      IF(J.NE.1) GOTO 68
      IF(J.EQ.1) GOTO 69
   69 L=(I-1)*12+J
      R(L)=TR
      GOTO 70
   68 L=(I-1)*12+J
      R(L)=0.0
   70 CONTINUE
   71 CONTINUE
      GOTO 79
C*****CONTINUOUS UNIFORM RECRUITMENT
   87 DO 85 I=1,NMT
      R(I)=TR/XDT
   85 CONTINUE
      GOTO 79
C*****TRIGONOMETRIC RECRUITMENT ****************************
C     VARIABLE NAME = PER                                 *
C     PER = 12 (BIMODAL ANNUALLY)                         *
C     PER = 24 (UNIMODAL ANNUALLY)                        *
C**********************************************************
   88 PI=3.1415927
      AMPL=(2*PI)/PER
      SUM=0.
      S2=0.
      S3=0.
      DO 188 J=1,NDT
      RA(J)=0.
      RA(J)=SIN(AMPL*J)
      RA(J)=ABS(RA(J))
      SUM=SUM+RA(J)
```

```
  188 CONTINUE
      DO 187 J=1,NDT
      RC(J)=0.
      RC(J)=RA(J)/SUM
      S2=S2+RC(J)
  187 CONTINUE
      LCOUNT=0
      DO 189 I=1,NTL
      S3=0.
      K=0
      DO 186 J=1,NDT
      K=LCOUNT+J
      R(K)=RC(J)*TR
      S3=S3+R(K)
  186 CONTINUE
      LCOUNT=LCOUNT+12
  189 CONTINUE
      IF(MOS.EQ.0) GOTO 79
      DO 184 J=NMC,NMT
      KK=J-NM
      R(J)=RC(KK)*TR
  184 CONTINUE
      K=0
      GOTO 79
C*********************************************************************
C     BETA DISTRIBUTION                                              *
C                                                                    *
C     PARAMETERS OF THE BETA DISTRIBUTION                            *
C     PARM1 = ALPHA 1                                                *
C     PARM2 = ALPHA 2                                                *
C*********************************************************************
   90 PI=3.1415927
      A1P=PARM1-1.
      A2P=PARM2-1.
      A3P=(PARM1+PARM2)-1.
      XN1F=EXP(-A1P)*(A1P**A1P)*(SQRT(2*PI*A1P))
      XN2F=EXP(-A2P)*(A2P**A2P)*(SQRT(2*PI*A2P))
      XN3F=EXP(-A3P)*(A3P**A3P)*(SQRT(2*PI*A3P))
      BETADS=(XN1F*XN2F)/XN3F
      SUM=0.
      S2=0.
      DO 220 J=1,NDT
      XXX=J
      TIME=XXX/XDT
      RA(J)=0.
      RA(J)=((TIME**A1P)*((1-TIME)**A2P))/BETADS
      SUM=SUM+RA(J)
  220 CONTINUE
      DO 219 J=1,NDT
      RC(J)=0.
      RC(J)=RA(J)/SUM
      S2=S2+RC(J)
  219 CONTINUE
      LCOUNT=0
      DO 218 I=1,NTL
```

```
      S3=0.
      K=0
      DO 217 J=1,NDT
      K=LCOUNT+J
      R(K)=RC(J)*TR
      S3=S3+R(K)
  217 CONTINUE
      LCOUNT=LCOUNT+12
      K=0
  218 CONTINUE
      IF(MOS.EQ.0) GOTO 79
      DO 216 J=NMC,NMT
      KK=J-NM
      R(J)=RC(KK)*TR
  216 CONTINUE
      GOTO 79
C***********************************************************************
C     FORWARD AND CENTERED SIMULATIONS                                 *
C***********************************************************************
C                                                                      *
C     CALCULATION OF CONTINUOUS RECRUITMENT LOOPS                      *
C                                                                      *
C***********************************************************************
   79 WRITE(*,80)
   80 FORMAT(7X,'CALCULATING POPULATION VECTOR(S):'//)
      SMOS=1.0
      JTRANS=0.
      JJJ=0.0
      DO 4000 K=1,NMT
      IF(K.GE.NMC) GOTO 999
      IF(K.LE.NM) GOTO 998
  999 SMOS=SMOS+1
  998 TOTLP=0.
      TOTLC=0.
      TOTWC=0.
      YWJ=0.
      YNJ=0.
      BMJ=0.
      RPJ=0.
      PJ=0.
      AVGL=0.
      EYPR=0.
      SSB=0.
      PPA=0.
      CXT=0.
      PJCX=0.
      IF(JTRANS.GT.0) GOTO 3998
      ABMJ=0.
      ASSB=0.
      AYNJ=0.
      ATOTLC=0.
      ATOTWC=0.
      APJ=0.
      DO 3999 JJ=1,NM
      AP(JJ)=0.0
```

```
      AYN(JJ)=0.0
      AYW(JJ)=0.0
 3999 CONTINUE
      JJJ=1
 3998 JTRANS=JTRANS+JJJ
      IF(K.LE.4) GOTO 3997
      IF(K.GT.4) GOTO 3996
 3996 DO 3995 KT=2,4
      X=K
      J=1
      DO 3994 I=X,SMOS,-1
      P(J,KT-1)=P(J,KT)
      J=J+1
 3994 CONTINUE
 3995 CONTINUE
      KL=4
      GOTO 3800
 3997 KL=K
 3800 X=K
      J=1
      DO 1000 I=X,SMOS,-1
      IF(I-X)110,120,110
  120 P(J,KL)=R(K)
      GOTO 150
  110 Z(J)=XM(J)+(A(J)*F(J))
      IF(K.EQ.1) GOTO 118
      IF(K.GT.1) GOTO 118
C*****FORWARD POPULATION DERIVATIVES
  118 P(J,KL)=P(J-1,KL-1)+((DT*(-Z(J)*P(J-1,KL-1)))*(1+XMU(1)))
      GOTO 22
C*****CENTERED POPULATION DERIVATIVES
  119 P(J,KL)=P(J-2,KL-2)+(((2*DT)*(-Z(J)*P(J-1,KL-1)))*(1+XMU(1)))
   22 Z1(J)=0.
      RN(J)=0.
      YN(J)=0.
      YW(J)=0.
      Z1(J)=(XM(J)/XDT)+(A(J)*(F(J)/XDT))
      IF(Z1(J).EQ.0) GOTO 111
      RN(J)=(P(J,KL)/Z1(J))*(1.0-EXP(-Z1(J)))
      RPJ=RPJ+RN(J)
      YN(J)=A(J)*(F(J)/XDT)*RN(J)
      YW(J)=YN(J)*WT(J)
      GOTO 83
  111 YN(J)=0.
      YW(J)=0.
   83 BIOM(J)=0.
C       POPP(J)=0.
      IF(J.LT.MFR) GOTO 112
      PJCX=PJCX+P(J,KL)
      BIOM(J)=WT(J)*RN(J)
C       POPP(J)=RN(J)*FEC(J)
C       PPA=PPA+POPP(J)
      BMJ=BMJ+BIOM(J)
      PJ=PJ+P(J,KL)
      AP(J)=AP(J)+P(J,KL)
```

```
      IF(J.LT.MRM) GOTO 112
      SSB=SSB+BIOM(J)
  112 YNJ=YNJ+YN(J)
      AYN(J)=AYN(J)+YN(J)
      YWJ=YWJ+YW(J)
      AYW(J)=AYW(J)+YW(J)
      TOTLC=TOTLC+(YN(J)*SLT(J))
  150 J=J+1
 1000 CONTINUE
      IF(K.LT.NM) GOTO 4000
      WRITE(*,1001)K
 1001 FORMAT(7X,'EQUILIBRIUM CONDITIONS SATISFIED IN PERIOD ',
     $I3,'...!'//)
      IF(YNJ)1430,3900,1430
 1430 ALC(K)=TOTLC/YNJ
      AVWT(K)=YWJ/YNJ
 3900 ABMJ=ABMJ+BMJ
      AYNJ=AYNJ+YNJ
      ASSB=ASSB+SSB
      ATOTLC=ATOTLC+TOTLC
      ATOTWC=ATOTWC+YWJ
      APJ=APJ+PJ
      IF(NSRCH)1003,1003,1002
C**********************************************************************
C     LENGTH @ AGE PROBABILITY DISTRIBUTIONS                          *
C**********************************************************************
 1002 WRITE(*,1004)K
 1004 FORMAT(7X,'CALCULATING PROBABILITY SEARCH FOR PERIOD ',I3,/,
     *7X,'SIT TIGHT...THIS MAY TAKE AWHILE!'/)
      X=K
      J=1
      DO 990 I=X,SMOS,-1
      DO 970 KK=1,NDIST
      IF(KK.LE.15) GOTO 968
      IF(KK.EQ.16) GOTO 967
      IF(KK.GT.16) GOTO 966
  968 XLD(J,KK)=SLT(J)-((16-KK)*0.2)*SE(J)*NAMP
C     PD(J,KK)=P(J,KL)*FRQ(KK)
      PDY(J,KK)=YN(J)*FRQ(KK)
      GOTO 970
  967 XLD(J,KK)=SLT(J)
C     PD(J,KK)=P(J,KL)*FRQ(KK)
      PDY(J,KK)=YN(J)*FRQ(KK)
      GOTO 970
  966 XLD(J,KK)=SLT(J)+((KK-16)*0.2)*SE(J)*NAMP
C     PD(J,KK)=P(J,KL)*FRQ(KK)
      PDY(J,KK)=YN(J)*FRQ(KK)
  970 CONTINUE
  980 CONTINUE
      J=J+1
  990 CONTINUE
      X=K
      J=1
C*****PLOTTING LENGTH @ AGE DISTRIBUTIONS
      DO 835 I=X,SMOS,-1
```

```
      DO 834 KK=1,NDIST
      XX=J
C      WRITE(7,833)XX,XLD(J,KK),K
C      WRITE(8,833)XLD(J,KK),PD(J,KK),K
C      WRITE(12,833)XLD(J,KK),PDY(J,KK),K
  833 FORMAT(F15.7,F15.7,I5)
  834 CONTINUE
      J=J+1
  835 CONTINUE
C**********************************************************************
C     SEARCHING ROUTINE FOR SCANNING                                  *
C     PROBABILISTIC LENGTH DISTRIBUTIONS                              *
C**********************************************************************
      TSY=0.
      XLMAX=0.
      XLMIN=0.
      NTOTLI=0.
      RANGEL=0.
      XMINLEN=0.
      XMAXLEN=0.
      XLMAX=SLT(NM)+50.0*SE(NM)
      XLMIN=SLT(MFR)-12.0*SE(MFR)
      XLMIN=INT(XLMIN)
      XLMAX=INT(XLMAX)
      IF(XLMIN.GE.0) GOTO 958
      XLMIN=0.0
  958 NTOTLI=INT((XLMAX-XLMIN)/DTL)
      RANGEL=XLMAX-XLMIN
      DO 955 ZZ=1,NTOTLI
C      POPLEN(ZZ)=0.
      SYLND(ZZ)=0.
      IF(ZZ.NE.1) GOTO 940
      XMINLEN=XLMIN
      XMAXLEN=XLMIN+DTL
      GOTO 942
  940 XMINLEN=XMAXLEN
      XMAXLEN=XMINLEN+DTL
  942 XLENPT(ZZ)=(XMAXLEN+XMINLEN)/2.
      X=K
      J=1
      DO 950 I=X,SMOS,-1
      DO 949 KK=1,NDIST
      IF(J.LT.MFR) GOTO 949
      IF(XLD(J,KK).LE.XMAXLEN) GOTO 947
      GOTO 949
  947 IF(XLD(J,KK).GE.XMINLEN) GOTO 945
      GOTO 949
  945 SYLND(ZZ)=SYLND(ZZ)+PDY(J,KK)
C     POPLEN(ZZ)=POPLEN(ZZ)+PD(J,KK)
  949 CONTINUE
      J=J+1
  950 CONTINUE
      TSY=TSY+SYLND(ZZ)
  955 CONTINUE
C**********************************************************************
```

```
C      SEARCHING ROUTINE FOR SCANNING DETERMINISTIC                         *
C      POPULATION LENGTH DISTRIBUTION                                       *
C***********************************************************************
C       XLMAX=0.
C       XLMIN=0.
C       NNTOTLI=0.
C       RANGEL=0.
C       XMINLEN=0.
C       XMAXLEN=0.
C       TYP=0.
C       TPD=0.
C       XLMAX=SLT(NM)+50.0*SE(NM)
C       XLMIN=SLT(MFR)-12.0*SE(MFR)
C       XLMIN=INT(XLMIN)
C       XLMAX=INT(XLMAX)
C       IF(XLMIN.GE.0) GOTO 1958
C       XLMIN=0.
C 1958  NNTOTLI=INT((XLMAX-XLMIN)/DTL)
C       RANGEL=XLMAX-XLMIN
C       DO 1955 ZZ=1,NNTOTLI
C       DPOP(ZZ)=0.
C       DYPD(ZZ)=0.
C       IF(ZZ.NE.1) GOTO 1940
C       XMINLEN=XLMIN
C       XMAXLEN=XLMIN+DTL
C       GOTO 1942
C 1940  XMINLEN=XMAXLEN
C       XMAXLEN=XMINLEN+DTL
C 1942  XDLENPT(ZZ)=(XMAXLEN+XMINLEN)/2.
C       X=K
C       J=1
C       DO 1950 I=X,SMOS,-1
C       IF(J.LT.MFR) GOTO 1949
C       IF(SLT(J).LE.XMAXLEN) GOTO 1947
C       GOTO 1949
C 1947  IF(SLT(J).GE.XMINLEN) GOTO 1945
C       GOTO 1949
C 1945  DPOP(ZZ)=DPOP(ZZ)+P(J,KL)
C       DYPD(ZZ)=DYPD(ZZ)+YN(J)
C 1949  J=J+1
C 1950  CONTINUE
C       TYD=TYD+DYPD(ZZ)
C       TPD=TPD+DPOP(ZZ)
C 1955  CONTINUE
C***********************************************************************
C      DYNAMIC LENGTH @ AGE PROBABILITY DISTRIBUTIONS                       *
C***********************************************************************
C***********************************************************************
C      WRITING TO DATA FILES                                                *
C***********************************************************************
 1003 X=K
      J=1
      DO 5820 I=X,SMOS,-1
      IF(J.LT.MFR) GOTO 5819
 5815 WRITE(4,5811)SLT(J),P(J,KL)
```

```
 5811 FORMAT(F15.5,F15.5)
C      WRITE(2,5813)SLT(J),POPP(J),NPLOT
C 5813 FORMAT(F15.4,F15.5,I5)
 5819 J=J+1
 5820 CONTINUE
C      DO 5830 J=1,NNTOTLI
C      WRITE(11,5828)XDLENPT(J),DPOP(J),K
C 5830 CONTINUE
 5828 FORMAT(F15.5,F15.5,I5)
      KKK=K+1
      DO 5899 J=1,NTOTLI
C      WRITE(11,5828)XDLENPT(J),DYPD(J),KKK
 5899 CONTINUE
      KKKK=K+200
      DO 5901 J=1,NTOTLI
C      WRITE(4,5828)XLENPT(J),POPLEN(J),K
      WRITE(12,5828)XLENPT(J),SYLND(J)
 5901 CONTINUE
C      WRITE(11,5836)TPD,TYD,TSY
C 5836 FORMAT(5X,'TOTAL POPN NUMBERS = ',E18.7,/5X,
C     $        'TOTAL YIELD IN NUMBERS = ',E18.7,/5X,
C     $        'TOTAL STOCHASTIC YIELD IN NUMBERS = ',E18.7)
      IF(K.LT.NM) GOTO 4000
      IF(K.GE.NM) GOTO 3901
C*********************************************************************
C     TRANSITIONAL OUTPUT MODULE                                     *
C*********************************************************************
 3901 MR=K-MFR
      IF(R(MR).EQ.0) GOTO 3902
      YPR=YWJ/R(MR)
      GOTO 3903
 3902 YPR=0.0
 3903 WRITE(5,1590)K,FMORT
 1590 FORMAT(7X,'MONTH = ',I5,10X,'ANNUAL F = ',F8.4,/)
      XNY=K-NM
C      WRITE(9,5811)XNY,ALC(K),NPLOT
C      WRITE(10,5811)XNY,YWJ,NPLOT
      IF(NOUT.EQ.0) GOTO 3008
      IF(NOUT.EQ.1) GOTO 1591
 1591 WRITE(5,1592)
 1592 FORMAT(1X,'AGE',8X,'POPN NO.',3X,'LENGTH',5X
     $'WEIGHT',7X,'BIOMASS',3X,'YIELD WT.',1X,'YIELD NO.')
      X=K
      J=1
      DO 1700 I=X,SMOS,-1
      WRITE(5,1640)J,P(J,KL),SLT(J),WT(J),BIOM(J),YW(J),YN(J),F(J)
 1640 FORMAT(1X,I3,3X,F13.3,2X,F8.2,2X,F8.3,2X,F13.4,2X,F9.4,2X,F8.4,
     $2X,F5.3)
      J=J+1
 1700 CONTINUE
      WRITE(5,1701)PJ,BMJ,YWJ,YNJ
 1701 FORMAT(/1X,'TOTAL',1X,E13.7,22X,E13.7,1X,E13.7,2X,E13.7)
      WRITE(5,1710)ALC(K),AVWT(K)
 1710 FORMAT(7X,
     $'AVG. LENGTH CATCH = ',F13.6,/,7X,'AVG. WT. = ',F13.6)
```

```
      WRITE(5,1712)YPR
 1712 FORMAT(7X,'YIELD PER RECRUIT = ',F15.8//)
      GOTO 3060
 3008 WRITE(5,3010)
 3010 FORMAT(7X,'CONDENSED TRANSITIONAL OUTPUT:',//)
      WRITE(5,3020)PJ,BMJ,SSB
 3020 FORMAT(7X,'POPULATION ABUNDANCE = ',F12.3,/,7X,
     $'POPULATION BIOMASS = ',E14.7,/,7X,'SPAWNING STOCK BIOMASS = ',
     $E14.7)
      WRITE(5,3030)YWJ,YNJ
 3030 FORMAT(7X,'YIELD IN WEIGHT = ',
     $E14.7,/,7X,'YIELD IN NUMBERS = ',E14.7)
      WRITE(5,3040)ALC(K)
 3040 FORMAT(7X,
     $'AVG. LENGTH IN CATCH = ',F8.4)
      WRITE(5,3050)AVWT(K)
 3050 FORMAT(7X,'AVG. WT. IN CATCH = ',F12.4)
      WRITE(5,3055)R(K),YPR
 3055 FORMAT(7X,'RECRUITMENT = ',F10.5,/,
     $7X,'YIELD PER RECRUIT = ',F9.4,//)
 3060 IF(JTRANS.NE.12) GOTO 4000
      IF(JTRANS.EQ.12) GOTO 3101
 3101 WRITE(5,3100)
 3100 FORMAT(7X,'AGE',8X,'POPN. NO.',3X,'Yweight',5X,'Ynumber')
      X=K
      J=1
      DO 3005 I=X,SMOS,-1
      WRITE(5,3004)J,AP(J),AYW(J),AYN(J),SLT(J)
 3004 FORMAT(7X,I3,2X,F10.2,5X,F12.3,5X,F12.3,5X,F12.3)
      J=J+1
 3005 CONTINUE
      WRITE(5,3110)ABMJ,ASSB,APJ
 3110 FORMAT(//,7X,'BIOMASS = ',E14.7,/,7X,'SPAWNING BIOMASS = ',
     $E14.7,/,7X,'POPULATION SIZE = ',E14.7)
      WRITE(5,3120)ATOTWC,AYNJ
 3120 FORMAT(7X,'YIELD IN WEIGHT = ',E14.7,//,7X,'YIELD IN NO. = ',
     $E14.7)
      IF(AYNJ.LE.0) GOTO 3128
      AALC=ATOTLC/AYNJ
      AAVWT=ATOTWC/AYNJ
      GOTO 3129
 3128 AALC=0.
      AAVWT=0.
      GOTO 3129
 3129 WRITE(5,3130)AALC,AAVWT
 3130 FORMAT(7X,'AVG. LENGTH IN CATCH = ',F8.4,5X,'AVG. WEIGHT IN
     $CATCH = ',F8.4)
      WRITE(5,3140)YPR
 3140 FORMAT(7X,'ANNUAL YIELD PER RECRUIT = ',F9.6,//)
      JTRANS=0.
      GOTO 4000
 4000 CONTINUE
 4001 END
```

```fortran
      PROGRAM CUREX
C*******************************************************
C                                                      *
C     AULT, JERALD S. 1987.                            *
C     PROGRAM CURECS. CODED IN VAX FORTRAN V4.5-219    *
C     PROGRAM TO SIMULATE A QUASI-CONTINUOUS RECRUITING *
C     LIFE HISTORY TYPICAL OF TROPICAL MARINE FISH STOCKS. *
C*******************************************************
      REAL*8 FRQ(40),SE(200),PD(160,35),POPLEN(2500),XLD(160,35),
     *       XLENPT(2500),POM(200),DPUP(2000),DYPD(2000),XDLENPT(2000),
     *       PDY(160,35),PDYLD(200),SYLND(2500),TSL(2500),TDL(2500)
      DIMENSION XM(200,200),A(200,200),F(200,200),P(600,200),
     *R(200),SLT(200),WT(200),AL(200),TL(200),Y(200),BIOM(200),
     *TW(200,200),RN(200,200),YN(200,200),XOMEG(4),FM(200),FIMF(200),
     *SXR(200),ALC(200),TLC(200),Z(200,200),XE(200),PA(200),RC(200),
     *AVWT(200),FEC(200),PBIRTH(200),XFE(200),WTF(200),CX(200),TWC(200),
     *AVWT2(200),SLT1(200),BIT(200),BIRTHS(200),ACHIR(200),RBIOM(20 ),
     *AP(200),AYW(200),AYN(200),FP(12),AM(12),POPF(200)
      OPEN(UNIT=12,NAME='STYLDN.DAT',TYPE='NEW',DISP='KEEP')
      OPEN(UNIT=11,NAME='LENFRFQ.DAT',TYPE='NEW',DISP='KEEP')
      OPEN(UNIT=10,NAME='YLDPR.DAT',TYPE='NEW',DISP='KEEP')
      OPEN(UNIT=9,NAME='AVLENGTH.DAT',TYPE='NEW',DISP='KEEP')
      OPEN(UNIT=8,NAME='PROBDEN.DAT',TYPE='NEW',DISP='KEEP')
      OPEN(UNIT=7,NAME='LENGTH.DAT',TYPE='NEW',DISP='KEEP')
      OPEN(UNIT=5,NAME='REPRO.DAT',TYPE='NEW',DISP='KEEP')
      OPEN(UNIT=4,NAME='LENDIST.DAT',TYPE='NEW',DISP='KEEP')
      OPEN(UNIT=3,NAME='FRAC.DAT',TYPE='NEW',DISP='KEEP')
      OPEN(UNIT=2,NAME='FEC.DAT',TYPE='NEW',DISP='KEEP')
      OPEN(UNIT=1,NAME='SCRUTO.DAT',TYPE='OLD',READONLY)
      WRITE(6,30)
   30 FORMAT(7X,'INITIALIZING CONTINUOUS TIME DATA ARRAYS',//)
      READ(1,31)NDT,MOS,NTL,MFR,MFC,MRM,NFOT,NROT,
     *          NOUT,NAVA,NLG,NOPT,NDIST,NAMP,NPLOT
      READ(1,29)ULINF,WINF,XAKV,TO,XNM,FMORT,ALPHA,BETA,
     *          EQUIL,TR,PER,PARM1,PARM2,XNMF,DTL,ALSLB
      READ(1,33)RSLB
      READ(1,29)FP(1),FP(2),FP(3),FP(4),FP(5),FP(5),FP(7),FP(8),
     *          FP(9),FP(10),FP(11),FP(12)
      READ(1,29)AM(1),AM(2),AM(3),AM(4),AM(5),AM(6),AM(7),AM(8),
     *          AM(9),AM(10),AM(11),AM(12)
      READ(1,29)(FRQ(I),I=1,NDIST)
   31 FORMAT(8I10)
   33 FORMAT(E10.4)
   29 FORMAT(8F10.0)
C***********************************************************
C     INITIALIZE MATRIX ARRAYS                             *
C***********************************************************
C**********************************************************
C                                                          *
C     NDT = FINITE DIVISION OF TIME STEP FOR SIMULATION   *
C     NM = NUMBER OF TIME STEPS OF SIMULATION TO          *
C          EQUILIBRIUM                                     *
C     MOS = TIME STEPS OF SIMULATION POST-EQUILIBRIUM     *
C     NTL = OLDEST TIME STEP IN THE CATCH                 *
C     MFC = TIME STEP OF FIRST CAPTURE                    *
C     MRM = TIME STEP OF REPRODUCTIVE MATURITY            *
C     MFR = TIME STEP OF RECRUITMENT TO THE EXPLOITABLE PHASE *
C     ULINF = ULTIMATE LENGTH - VON BERTALANFFY FORMULATION *
C     WINF = ULTIMATE WEIGHT  -  "          "              *
C     TO   = TIME AT WHICH WEIGHT/LENGTH OF FISH = ZERO   *
C     XNM = ANNUAL NATURAL MORTALITY RATE                 *
C     FMORT = ANNUAL FISHING MORTALITY RATE               *
C     ALPHA = A-COEFFICIENT FROM FECUNDITY ON WEIGHT RELATION *
C     BETA =  B-     "       "      "      "     "        *
C     XNMF = NUMBER OF MONTHS OF SEASONAL FISHING         *
C     FP(J) = SEASONAL FISHING MORTALITY SWITCH (J=1,...,12) *
C     XOMEG(T) = COUNTER FROM ANALYTICAL YIELD FORMULATION *
C     EQUIL = STATIONARY OR DYNAMIC TIME VECTORS (0=STAT,1=DYN) *
C                                                          *
C**********************************************************
      XOMEG(1)=1.0
      XOMEG(2)=-3.0
      XOMEG(3)=3.0
      XOMEG(4)=-1.0
      XDT=NDT
      NM=NTL*NDT
      NMC=NM+1
      NMT=NM+MOS
      XAKV=XAKV/XDT
```

```
      TO=10*XDT
      DO 55 I=1,NM
      SE(I)=ALSLB*((I)**BSLB)
   55 CONTINUE
C*******************************************************************
C     NATURAL MORTALITY ARRAY
C******************************************************************
C                                                                 *
C*****************************************************************
      DO 13 I = 1,NMT
      DO 12 J = 1,(MFR-1)
      XM(I,J)=0.0
   12 CONTINUE
   13 CONTINUE
C       DO 34 I = 1,NMT
C       XM(I,45)=(XNM/XDT)*0.36
C   34 CONTINUE
      DO 40 I = 1,NMT
      DO 35 J = MFR,NMT
      XM(I,J)=XNM/XDT
   35 CONTINUE
   40 CONTINUE
      GOTO 92
C*******************************************************************
C     AVAILABILITY ARRAY
C******************************************************************
C                                                                 *
C     SELECTIVITY PATTERN                                         *
C     VARIABLE NAME = NAVA                                        *
C     CONSTANT BY INTERVAL = 1                                    *
C     SEASONALLY AVAILABLE = 2                                    *
C                                                                 *
C******************************************************************
   92 IF(NAVA.EQ.1) GOTO 41
      IF(NAVA.EQ.2) GOTO 91
   41 DO 43 I = 1,NMT
      DO 42 J = 1,(MFC-1)
      A(I,J)=0.0
   42 CONTINUE
   43 CONTINUE
      DO 50 I = 1,NMT
      DO 45 J = MFC,NM
      A(I,J)=1.0
   45 CONTINUE
   50 CONTINUE
      GOTO 200
C*******************************************************************
C     SEASONAL AVAILABILITY
C******************************************************************
   91 DO 107 I = 1,NLL
      DO 106 M = 1,12
      K=(I-1)*12+M
      IF(AM(M).EQ.0) GOTO 95
      IF(AM(M).EQ.1) GOTO 97
   95 X=K
      J=1
      DO 96 JJ = X,1,-1
      IF(J.GT.NMT) GOTO 96
      A(JJ,J)=0.0
      J=J+1
   96 CONTINUE
      GOTO 106
   97 X=K
      J=1
      DO 98 JJ = X,1,-1
      IF(J.GT.NMT) GOTO 98
      A(JJ,J)=1.0
      J=J+1
   98 CONTINUE
  106 CONTINUE
  107 CONTINUE
      GOTO 200
C*******************************************************************
C     BIRTH RATE FUNCTION ARRAY
C******************************************************************
C                                                                 *
C     CALCULATION OF AGE/SIZE SPECIFIC BIRTH RATES                *
C     (NET MATERNITY LXMX) BY MANIPULATION OF THE                 *
C     FECUNDITY ON WEIGHT RELATIONSHIP.                           *
```

```
C*****************************************************************
  200 RHO=0.8
      XLAMBDA=0.8
      FECUN=0.0
      SMPFEC=0.0
      DO 201 JJ=1,MRM
      FEC(JJ)=0.0
  201 CONTINUE
      DO 206 JJ=MRM,NM
      XEF(JJ)=1.-EXP(-XAKV*((JJ-1.0)-T0))
      WTF(JJ)=WINF*(XEF(JJ)**3.)
      FEC(JJ)=ALPHA*(WTF(JJ)**BETA)
      IF(RHO.NE.XLAMBDA) GOTO 202
      IF(RHO.EQ.XLAMBDA) GOTO 203
  202 BIT(JJ)=FEC(JJ)*(((XLAMBDA**(JJ+1-MRM))-(RHO**
     $(JJ+1-MRM)))/(RHO-XLAMBDA))
      GOTO 205
  203 BIT(JJ)=FEC(JJ)*(JJ+1-MRM)*(RHO**(JJ-MRM))
      GOTO 205
  205 BIPIHS(JJ)=BIT(JJ)*((EXP(-XM(1,JJ)))**(JJ-1))
      FECUN=FECUN+FEC(JJ)
  206 CONTINUE
      DO 207 JJ=1,NM
      PBIRTH(JJ)=FEC(JJ)/FECUN
      PBIRTH(JJ)=PBIRTH(JJ)*100.
      SMPFEC=SMPFEC+PBIRTH(JJ)
  207 CONTINUE
C**************************************************************
C     FISHING MORTALITY ARRAY                                 *
C**************************************************************
C                                                             *
C     FISHING MORTALITY PATTERN                               *
C     VARIABLE NAME = NFOT                                    *
C     POPE'S (1971) METHOD ASSUMPTION = 1                     *
C     CONSTANT BY INTERVAL = 2                                *
C     SEASONAL FISHING STRATEGY = 3                           *
C                                                             *
C**************************************************************
      IF(NFOT.EQ.1) GOTO 21
      IF(NFOT.EQ.2) GOTO 60
      IF(NFOT.EQ.3) GOTO 130
C**************************************************************
C     POPE'S MID-YEAR APPROXIMATION                           *
C**************************************************************
   21 DO 20 I=1,NMT
      DO 19 L=1,14
      L1=0
      L1=(L-1)*12
      IF(I.EQ.L1) GOTO 18
      IF(I.NE.L1) GOTO 19
   18 DO 17 J = 1,NMT
      F(I,J)=FMORT/XDT
   17 CONTINUE
      GOTO 20
   19 CONTINUE
      DO 15 J = 1,NMT
      F(I,J)=0.0
   15 CONTINUE
   20 CONTINUE
      GOTO 84
C*****************************************************************
C     F CONSTANT EVERY INTERVAL                                  *
C*****************************************************************
   60 DO 62 I = 1,NMT
      DO 61 J = 1,(MFC-1)
      F(I,J)=0.0
   61 CONTINUE
   62 CONTINUE
C     DO 63 I = 1,NMT
C     F(I,45)=(FMORT/XDT)*0.36
C  63 CONTINUE
      DO 67 I = 1,NMT
      DO 64 J = MFC,NMT
      F(I,J)=FMORT/XDT
   64 CONTINUE
   67 CONTINUE
      GOTO 84
C*****************************************************************
C     SEASONAL FISHING MORTALITY STRATEGY                        *
```

```
C*******************************************************
  130 DO 149 I = 1,NLG
      DO 148 M = 1,12
      K=(I-1)*12+M
      IF(FP(M).EQ.0) GOTO 131
      IF(FP(M).EQ.1) GOTO 135
  131 X=K
      J=1
      DO 141 JJ = X,1,-1
      IF(J.GT.NMT) GOTO 141
      F(JJ,J)=0.0
      J=J+1
  141 CONTINUE
      GOTO 148
  135 X=K
      J=1
      DO 142 JJ = X,1,-1
      IF(J.GT.NMT) GOTO 142
      F(JJ,J)=FMORT/XNMF
      J=J+1
  142 CONTINUE
  148 CONTINUE
  149 CONTINUE
      GOTO 84
C*******************************************************
C     RECRUITMENT ARRAY                                  *
C*******************************************************
C                                                        *
C     RECRUITMENT SEQUENCE OPTION                        *
C     VARIABLE NAME = NROT                               *
C     DISCRETE = 1                                       *
C     CONSTANT = 2                                       *
C     PERIODIC DISTRIBUTION = 3                          *
C     BETA DISTRIBUTION = 4                              *
C                                                        *
C     TR = TOTAL ANNUAL RECRUITMENT                      *
C                                                        *
C*******************************************************
   84 IF(NROT.EQ.1) GOTO 86
      IF(NROT.EQ.2) GOTO 87
      IF(NROT.EQ.3) GOTO 88
      IF(NROT.EQ.4) GOTO 90
C*******************************************
C     DISCRETE RECRUITMENT                 *
C*******************************************
   86 DO 71 I=1,13
      DO 70 J=1,12
      IF (J.NE.1) GO TO 68
      IF (J.EQ.1) GO TO 69
   69 L=(I-1)*12+J
      R(L)=TR
      GO TO 70
   68 L=(I-1)*12+J
      R(L)=0.0
   70 CONTINUE
   71 CONTINUE
      GO TO 79
C*********************************************
C     TIME CONSTANT RECRUITMENT              *
C*********************************************
   87 DO 85 I=1,NMT
      R(I)=TR/XDT
   85 CONTINUE
      GO TO 79
C*********************************************
C     PERIODIC RECRUITMENT                   *
C     VARIABLE NAME = PER                    *
C     PER = 12 (BIMODAL ANNUALLY)            *
C     PER = 24 (UNIMODAL ANNUALLY)           *
C*********************************************
   88 PI=3.1415927
      AMPL=2*PI/PER
      SUM=0.0
      S2=0.0
      S3=0.0
      DO 188 J=1,12
      RA(J)=SIN(AMPL*J)
      RA(J)=ABS(RA(J))
      SUM=SUM+RA(J)
```

32

```
  188 CONTINUE
      DO 187 J = 1,12
      RC(J)=RA(J)/SUM
      S2=S2+RC(J)
  187 CONTINUE
      LCOUNT=0.
      DO 189 I=1,NTL
      S3=0.
      K=0.
      DO 186 J=1,12
      K=LCOUNT+J
      R(K)=RC(J)*TR
      S3=S3+R(K)
  186 CONTINUE
      LCOUNT=LCOUNT+12
  189 CONTINUE
      IF(MOS.EQ.0) GOTO 79
      DO 184 J=NMC,NMT
      KK=J-NM
      R(J)=RC(KK)*TR
  184 CONTINUE
      K=0.
      GOTO 79
C*********************************************************************
C       BETA DISTRIBUTION                                            *
C                                                                    *
C       INPUT PARAMETERS OF THE BETA DISTRIBUTION.                   *
C       PARM1 = ALPHA 1                                              *
C       PARM2 = ALPHA 2                                              *
C                                                                    *
C*********************************************************************
   90 PI=3.1415927
      A1P=PARM1-1.
      A2P=PARM2-1.
      A3P=(PARM1+PARM2)-1.
      XN1F=EXP(-A1P)*(A1P**A1P)*(SQRT(2*PI*A1P))
      XN2F=EXP(-A2P)*(A2P**A2P)*(SQRT(2*PI*A2P))
      XN3F=EXP(-A3P)*(A3P**A3P)*(SQRT(2*PI*A3P))
      BETADS=(XN1F*XN2F)/XN3F
      SUM=0.0
      S2=0.0
      DO 220 J=1,12
      TIME=J/12.
      RA(J)=((TIME**A1P)*((1-TIME)**A2P))/BETADS
      SUM=SUM+RA(J)
  220 CONTINUE
      DO 219 J=1,12
      RC(J)=RA(J)/SUM
      S2=S2+RC(J)
  219 CONTINUE
      LCOUNT=0.
      DO 218 I=1,NTL
      S3=0.0
      K=0.
      DO 217 J=1,12
      K=LCOUNT+J
      R(K)=RC(J)*TR
      S3=S3+R(K)
  217 CONTINUE
      LCOUNT=LCOUNT+12
      K=0.
  218 CONTINUE
      IF(MOS.EQ.0) GOTO 79
      DO 216 J=NMC,NMT
      KK=J-NM
      R(J)=RC(KK)*TR
  216 CONTINUE
      GOTO 79
C*********************************************************************
C       STATIONARY CONDITIONS
C********************************************************************* 
C                                                                    *
C       CALCULATION OF CONTINUOUS RECRUITMENT LOOP                   *
C       FOR GENERATING EQUILIBRIUM CONDITIONS                        *
C                                                                    *
C*********************************************************************
   79 WRITE(6,80)
   80 FORMAT (7X,'CALCULATING EQUILIBRIUM POPULATION VECTOR:'//)
      DO 1000 K = 1,NM
```

```
      TOTLP=0.0
      TOTLC=0.0
      TOTWC=0.0
      YWJ=0.0
      YNJ=0.0
      BMJ=0.0
      RPJ=0.0
      PJ=0.0
      AVGL=0.0
      EYPR=0.0
      SSB=0.0
      PPA=0.0
      X=K
      J=1
      DO 500 I = X,1,-1
      IF(I-X)110,120,110
  120 P(I,J)=R(K)
      P(I,J+1)=P(I,J)*(EXP(-Z(I,J)))
      GOTO 150
  110 Z(I,J)=XM(I,J)+A(I,J)*F(I,J)
      P(I,J)=P(I,J-1)*EXP(-Z(I,J-1))
      XE(J)=1.-EXP(-XAKV*((J+0.5)-T0))
      IF(XE(J)) 81,81,82
   82 SLT(J)=ULINF*XE(J)
      WT(J)=WINF*(XE(J)**3.)
      GOTO 89
   81 SLT(J)=0.0
      WT(J)=0.0
   89 IF(Z(I,J).EQ.0.0) GOTO 111
      RN(I,J)=(P(I,J)/Z(I,J))*(1.0-EXP(-Z(I,J)))
      PPJ=RPJ+RN(I,J)
      YN(I,J)=A(I,J)*F(I,J)*RN(I,J)
      GOTO 83
  111 YN(I,J)=0.0
   83 IF(J.LT.MFR) GOTO 112
      BIOM(J)=WT(J)*RN(I,J)
      PUPP(J)=RN(I,J)*FEC(J)
      POPP(J)=PUPP(J)/100000.
      PPA=PPA+PUPP(J)
      BMJ=BMJ+BIOM(J)
      PJ=PJ+P(I,J)
      IF(J.LT.MRM) GOTO 112
      SSB=SSB+BIOM(J)
  112 YNJ=YNJ+YN(I,J)
      RPJ=RPJ+RN(I,J)
      TLC(J)=YN(I,J)*SLT(J)
      TOTLC=TOTLC+TLC(J)
      IF(J.LT.MFR) GOTO 108
      TL(J)=RN(I,J)*SLT(J)
      TOTLP=TOTLP+TL(J)
  108 TWC(J)=YN(I,J)*WT(J)
      TOTWC=TOTWC+TWC(J)
      VSUM=0.0
      DO 109 IV=1,4
      VN=IV-1
      IF(Z(I,J).EQ.0.) GOTO 109
      VSUM=VSUM+XOMEG(IV)*EXP(-VN*XAKV*((J-1)-T0))/
     S(Z(I,J)+VN*XAKV)*(1-EXP(-(Z(I,J)+VN*XAKV)))
  109 CONTINUE
      YW(I,J)=A(I,J)*F(I,J)*P(I,J)*WINF*VSUM
      YWJ=YWJ+YW(I,J)
  150 J=J+1
  500 CONTINUE
      IF (RPJ) 430,1000,430
  430 AL(K)=TOTLP/RPJ
      IF(YNJ)431,1000,431
  431 ALC(K)=TOTLC/YNJ
      AVWT(K)=YWJ/YNJ
      AVWT2(K)=TOTWC/YNJ
 1000 CONTINUE
      K=NM
      X=K
      J=1
      CXT=0.0
      PJCX=0.0
      DO 799 I=X,1,-1
      IF(J.LT.MFR) GOTO 798
      PJCX=PJCX+P(I,J)
  798 J=J+1
```

```
  799 CONTINUE
      J=1
      DO 810 I=X,1,-1
      IF(J.LT.MFR) GOTO 802
      CX(J)=P(I,J)/PJCX
      CX(J)=CX(J)*100.
      ACBIR(J)=PBIRTH(J)*P(I,J)*(TR/12.)
      GOTO 803
  802 CX(J)=0.0
      ACBI=0.0
  803 CXT=CXT+CX(J)/100.
      ACBT=ACBT+ACBIR(J)
      J=J+1
  810 CONTINUE
C***********************************************************************
C     LENGTH & AGE PROBABILITY DISTRIBUITONS                           *
C***********************************************************************
      K=NM
      X=K
      J=1
      DO 990 I=X,1,-1
      PDM(J)=P(1,J)
      PDYLD(J)=YN(I,J)
      DO 970 KK=1,NDIST
      IF(KK.LE.15) GOTO 968
      IF(KK.EQ.16) GOTO 967
      IF(KK.GT.16) GOTO 966
  968 XLD(J,KK)=SLT(J)-((16-KK)*0.2)*SE(J)*NAMP
      PD(J,KK)=PDM(J)*FRQ(KK)
      PDY(J,KK)=PDYLD(J)*FRQ(KK)
      GOTO 970
  967 XLD(J,KK)=SLT(J)
      PD(J,KK)=PDM(J)*FRQ(KK)
      PDY(J,KK)=PDYLD(J)*FRQ(KK)
      GOTO 970
  966 XLD(J,KK)=SLT(J)+((KK-16)*0.2)*SE(J)*NAMP
      PD(J,KK)=PDM(J)*FRQ(KK)
      PDY(J,KK)=PDYLD(J)*FRQ(KK)
  970 CONTINUE
  980 CONTINUE
      J=J+1
  990 CONTINUE
      X=K
      J=1
C*****PLOTTING LENGTH-&-AGE-DISTRIBUTIONS
      DO 835 I=X,1,-1
      DO 834 KK=1,NDIST
      XX=J
C      WRITE(7,833)XX,XLD(J,KK),K
C      WRITE(8,833)XLD(J,KK),PD(J,KK),K
C      WRITE(12,833)XLD(J,KK),PDY(J,KK),K
  833 FORMAT(F15.7,F15.7,I5)
  834 CONTINUE
      J=J+1
  835 CONTINUE
C***********************************************************************
C      SEARCHING ROUTINE FOR SCANNING PROBABILISTIC                    *
C      LENGTH DISTRIBUITONS                                            *
C***********************************************************************
      TSY=0.
      TSLC=0.
      SALC=0.
      XLMAX=SLT(NM)+50.0*SE(NM)
      XLMIN=SLT(MFR)-12.0*SE(MFR)
      XLMIN=INT(XLMIN)
      XLMAX=INT(XLMAX)
      IF(XLMIN.GE.0) GOTO 958
      XLMIN=0.0
  958 NTOTLI=INT((XLMAX-XLMIN)/DTL)
      RANGEL=XLMAX-XLMIN
      DO 955 ZZ=1,NTOTLI
      POPLEN(ZZ)=0.
      SYLND(ZZ)=0.
      TSL(ZZ)=0.
      IF(ZZ.NE.1) GOTO 940
      XMINLEN=XLMIN
      XMAXLEN=XLMIN+DTL
      GOTO 942
  940 XMINLEN=XMAXLEN
```

```
      XMAXLEN=XMINLEN+DTL
  942 XLENPT(ZZ)=(XMAXLEN+XMINLEN)/2.
      X=NM
      J=1
      DO 950 I=X,1,-1
      DO 949 KK=1,NDIST
      IF(J.LT.MFR) GOTO 949
      IF(XLD(J,KK).LE.XMAXLEN) GOTO 947
      GOTO 949
  947 IF(XLD(J,KK).GE.XMINLEN) GOTO 945
      GOTO 949
  945 POPLEN(ZZ)=POPLEN(ZZ)+PD(J,KK)
      SYLND(ZZ)=SYLND(ZZ)+PDY(J,KK)
      TSL(ZZ)=TSL(ZZ)+(PDY(J,KK)*XLENPT(ZZ))
  949 CONTINUE
      J=J+1
  950 CONTINUE
      TSY=TSY+SYLND(ZZ)
      TSLC=TSLC+TSL(ZZ)
  955 CONTINUE
      SALC=TSLC/TSY
C**********************************************************************
C     SEARCHING ROUTINE FOR SCANNING DETERMINISTIC                    *
C     POPULATION LENGTH DISTRIBUTIONS                                 *
C**********************************************************************
      XLMAX=0.
      XLMIN=0.
      NTOTL1=0
      RANGEL=0.
      XMINLEN=0.
      XMAXLEN=0.
      TYP=0.
      TPD=0.
      TDLC=0.
      DALC=0.
      TYD=0.
      XLMAX=SLT(NM)+50.0*SE(NM)
      XLMIN=SLT(MFR)-12.0*SE(MFR)
      XLMIN=INT(XLMIN)
      XLMAX=INT(XLMAX)
      IF(XLMIN.GE.0) GOTO 1958
      XLMIN=0.
 1958 NTOTL1=INT((XLMAX-XLMIN)/DTL)
      RANGEL=XLMAX-XLMIN
      DO 1955 ZZ=1,NTOTL1
      DPOP(ZZ)=0.
      DYPD(ZZ)=0.
      TDL(ZZ)=0.
      IF(ZZ.NE.1) GOTO 1940
      XMINLEN=XLMIN
      XMAXLEN=XLMIN+DTL
      GOTO 1942
 1940 XMINLEN=XMAXLEN
      XMAXLEN=XMINLEN+DTL
 1942 XDLENPT(ZZ)=(XMAXLEN+XMINLEN)/2.
      X=NM
      J=1
      DO 1950 I=X,1,-1
      IF(J.LT.MFR) GOTO 1949
      IF(SLT(J).LE.XMAXLEN) GOTO 1947
      GOTO 1949
 1947 IF(SLT(J).GE.XMINLEN) GOTO 1945
      GOTO 1949
 1945 DPOP(ZZ)=DPOP(ZZ)+P(I,J)
      DYPD(ZZ)=DYPD(ZZ)+YN(I,J)
      TDL(ZZ)=TDL(ZZ)+(YN(I,J)*XDLENPT(ZZ))
 1949 J=J+1
 1950 CONTINUE
      TYD=TYD+DYPD(ZZ)
      TPD=TPD+DPOP(ZZ)
      TDLC=TDLC+TDL(ZZ)
 1955 CONTINUE
      DALC=TDLC/TYD
C**********************************************************************
C     WRITING TO DATA FILES                                           *
C**********************************************************************
      K=NM
      X=K
      J=1
```

36

```
      DO 820 I=X,1,-1
      XXXX=J
      IF(J.LT.MFR) GOTO 819
  815 WRITE(3,811)SLT(J),YN(I,J),K
  811 FORMAT(F15.5,F15.5,I5)
C***        WRITE(2,813)SLT(J),PD(J),NPLOT
  813 FORMAT(F15.4,F15.5,I5)
  819 J=J+1
  820 CONTINUE
      DO 830 J=1,NTOTLT
C      WRITE(4,828)XLENPT(J),POPLEN(J),K
C      WRITE(11,828)XDLENPT(J),DPOP(J),K
  830 CONTINUE
  828 FORMAT(F15.5,F15.5,I5)
      KKK=NM+100
      DO 899 J=1,NTOTLT
      WRITE(11,828)XDLENPT(J),DYPD(J),KKK
  899 CONTINUE
C     WRITE(11,848)TDLC,TYD,DALC
  848 FORMAT(7X,'TDLC = ',E12.6,5X,'TYD = ',E12.6,/,
     $7X,'DETERMINISTICALLY AGGREGATED
     $ AVG LENGTH IN CATCH = ',F15.5)
      KKKK=NM+200
      DO 901 J=1,NTOTLT
      WRITE(12,828)XLENPT(J),SYLND(J),KKKK
  901 CONTINUE
C     WRITE(12,849)TSLC,TSY,SALC
  849 FORMAT(7X,'TSLC = ',E12.6,5X,'TSY = ',F12.6,/,
     $7X,'STOCHASTICALLY AGGREGATED
     $ AVG LENGTH IN CATCH = ',F15.5)
      WRITE(11,836)TPD,TYD,TSY
  836 FORMAT(5X,'TOTAL POPN NUMBERS = ',E18.7,/5X,
     $      'TOTAL YIELD IN NUMBERS = ',E18.7,/5X,
     $      'TOTAL STOCHASTIC YIELD IN NUMBERS = ',E18.7)
C*********************************************************************
C                                                                   *
C            EQUILIBRIUM OUTPUT MODULE                              *
C                                                                   *
C*********************************************************************
      K=NM
      MR=NM-MFR
      IF(R(K).EQ.0) GOTO 840
      EYPR=YWJ/R(MR)
  840 EYPR=0.0
      WRITE(5,590)K,FMORT
  590 FORMAT(7X,'MONTH = ',I5,10X,'ANNUAL F = ',F8.4,/)
      WRITE(5,591)
  591 FORMAT(1X,'AGE',8X,'POPN NO.',5X,'LENGTH',5X,
     *'WEIGHT',7X,'BIOMASS',3X,'YIELD WT.',1X,'YIELD NO.')
      WRITE(6,600)K,FMORT
  600 FORMAT(7X,'MONTH = ',I5,10X,'ANNUAL F = ',F8.4,/)
      WRITE(6,610)
  610 FORMAT(1X,'AGE',8X,'POPN NO.',4X,'LENGTH',5X,
     *'WEIGHT',7X,'BIOMASS',3X,'YIELD WT.',1X,'YIELD NO.')
      X=K
      J=1
      DO 700 I = X,1,-1
     /WRITE(5,640)J,P(I,J),SLT(J),WT(J),BIOM(J),YW(I,J),YN(I,J),F(I,J)
  640 FORMAT(1X,I3,3X,F13.3,2X,F10.5,2X,F8.3,2X,F13.4,2X,F9.4,2X,F8.4,
     $2X,F5.3)
      WRITE (6,650)J,P(I,J),SLT(J),WT(J),BIOM(J),YW(I,J),YN(I,J),F(I,J)
  650 FORMAT(1X,I3,3X,F13.3,2X,F8.3,2X,F8.3,2X,F13.4,2X,F9.4,2X,F8.4,
     $2X,F5.3)
      J=J+1
  700 CONTINUE
      WRITE(5,701)PJ,BMJ,YWJ,YNJ
  701 FORMAT(/1X,'TOTAL',1X,E13.7,22X,E13.7,1X,E13.7,2X,E13.7)
      WRITE(6,702)PJ,BMJ,YWJ,YNJ
  702 FORMAT(/1X,'TOTAL',1X,E13.7,22X,E13.7,1X,E13.7,2X,E13.7)
      WRITE(5,710)AL(K),ALC(K),AVWT(K)
  710 FORMAT(7X,'AVG. LENGTH POPN = ',F13.5,2X,
     $'AVG. LENGTH CATCH = ',F15.7,/,7X,'AVG. WT. = ',F13.6)
      WRITE(6,711)AL(K),ALC(K),AVWT(K),AVWT2(K)
  711 FORMAT(7X,'AVG. LENGTH POPN = ',F13.5,2X,
     $'AVG. LENGTH CATCH = ',F15.7,/,7X,'AVG. WT = ',F13.6,10X,'AVG.WT
     $ METHOD2 = ',F13.6,/)
      WRITE(5,712)EYPR,SSB
  712 FORMAT(7X,'EQUILIBRIUM YIELD PER RECRUIT = ',F15.8,
     $/,7X,'SPAWNING STOCK BIOMASS = ',F15.5//)
```

```
      WRITE(6,716)EYPR,SSB,BMJ
  716 FORMAT(7X,'EQUILIBRIUM YIELD PER RECRUIT = ',F15.8,
     $/,7X,'SPAWNING STOCK BIOMASS = ',F15.5,/,7X,
     $'TOTAL STOCK BIOMASS = ',F15.5//)
      WRITE(5,713)
  713 FORMAT(//,14X,'FECUNDITY(JTH)',8X,'P(BIRTH) ',10X,'RECRUITS(J)',
     $9X,'FRACTION CX(J)')
      X=K
      J=1
      DO 720 I = X,1,-1
      WRITE(5,714)J,FEC(J),BIT(J),BIRTHS(J),CX(J)
  714 FORMAT(7X,I5,2X,E14.7,6X,E14.7,6X,E14.7,6X,E12.6)
      J=J+1
  720 CONTINUE
      XNY=K-NM
      WRITE(9,730)XNY,ALC(K),NPLOT
      WRITE(10,730)XNY,YWJ,NPLOT
  730 FORMAT(F15.4,F15.5,I5)
      WRITE(5,715)FECUN,SMPFEC,ACBT,CXT
  715 FORMAT(/,14X,E14.7,6X,E14.7,6X,E14.7,6X,E12.6//)
      IF(EQUIL.EQ.0) GOTO 4001
      IF(EQUIL.EQ.1) GOTO 1501
C****************************************************************
C       DYNAMIC CONDITIONS
C***************************************************************
C                                                              *
C       CALCULATION OF CONTINUOUS RECRUITMENT LOOP             *
C       FOR GENERATING TRANSITIONAL CONDITIONS                 *
C                                                              *
C****************************************************************
 1501 WRITE(6,1502)
 1502 FORMAT (7X,' CALCULATING TRANSITIONAL POPULATION VECTORS:'//)
      SMOS=1.
      DO 4000 K = NMC,NMT
      SMOS=SMOS+1
      TOTLP=0.0
      TOTLC=0.0
      TOTWC=0.0
      YWJ=0.0
      YNJ=0.0
      BMJ=0.0
      SSB=0.0
      PJ=0.0
      AVGL=0.0
      EYPR=0.0
      IF(JTRANS.GT.0) GOTO 3998
      ABMJ=0.0
      ASSB=0.0
      AYNJ=0.0
      ATOTLC=0.0
      ATOTLP=0.0
      ATOTWC=0.0
      APJ=0.0
      DO 3999 JJ=1,NM
      AP(JJ)=0.0
      AYN(JJ)=0.0
      AYN(JJ)=0.0
 3999 CONTINUE
      JJJ=1
 3998 JTRANS=JTRANS+JJJ
      X=K
      J=1
      DO 1600 I = X,SMOS,-1
      IF(I-X)1610,1620,1610
 1620 P(I,J)=R(K)
      GOTO 1550
 1610 Z(I,J)=XM(I,J)+A(I,J)*F(I,J)
      P(I,J)=P(I,J-1)*EXP(-Z(I,J-1))
      XE(J)=1.-EXP(-XAKV*((J+0.5)-T0))
      IF(XE(J))1581,1581,1582
 1582 SLT(J)=ULINF*XE(J)
      WT(J)=WINF*(XE(J)**3.)
      GOTO 1583
 1581 SLT(J)=0.0
      WT(J)=0.0
 1583 BIOM(J)=WT(J)*P(I,J)
      BMJ=BMJ+BIOM(J)
      PJ=PJ+P(I,J)
      TL(J)=RN(I,J)*SLT(J)
```

38

```
      TOTLP=TOTLP+PS(J)
      AP(J)=AP(J)+P(I,J)
      IF(J.LT.MRM) GOTO 1515
      SSB=SSB+BIOM(J)
 1515 IF(Z(I,J).EQ.0.0) GOTO 1511
      RN(I,J)=(P(I,J)/Z(I,J))*(1.0-EXP(-Z(I,J)))
      YN(I,J)=A(I,J)*F(I,J)*RN(I,J)
      GOTO 1512
 1511 YN(I,J)=0.0
 1512 YNJ=YNJ+YN(I,J)
      AYN(J)=AYN(J)+YN(I,J)
      TWC(J)=YN(I,J)*WT(J)
      TOTWC=TOTWC+TWC(J)
      TLC(J)=YN(I,J)*SLT(J)
      TOTLC=TOTLC+TLC(J)
      VSUM=0.0
      DO 1509 IV=1,4
      VN=IV-1
      IF(Z(I,J).EQ.0.) GOTO 1509
      VSUM=VSUM+XOMEG(IV)*EXP(-VN*XAKV*((J-1)-T0))/
     S(Z(I,J)+VN*XAKV)*(1-EXP(-(Z(I,J)+VN*XAKV)))
 1509 CONTINUE
      YW(I,J)=A(I,J)*F(I,J)*P(I,J)*WINF*VSUM
      YWJ=YWJ+YW(I,J)
      AYW(J)=AYW(J)+YW(I,J)
 1550 J=J+1
 1600 CONTINUE
      IF (PJ)1430,3900,1430
 1430 AL(K)=TOTLP/PJ
      IF(YNJ)1431,3900,1431
 1431 ALC(K)=TOTLC/YNJ
      AVWT(K)=YWJ/YNJ
      AVWT2(K)=TOTWC/YNJ
 3900 ABMJ=ABMJ+BMJ
      AYNJ=AYNJ+YNJ
      ASSB=ASSB+SSB
      ATOTLC=ATOTLC+TOTLC
      ATOTLP=ATOTLP+TOTLP
      ATOTWC=ATOTWC+TOTWC
      APJ=APJ+PJ
      X=K
      J=1
      DO 825 I=X,SMOS,-1
      IF(J.LT.MFR) GOTO 824
      WRITE(3,821)SLT(J),YN(I,J),K
  821 FORMAT(F15.4,F15.5,I5)
  824 J=J+1
  825 CONTINUE
C*****************************************************************
C     DYNAMIC LENGTH & AGE PROBABILITY DISTRIBUTIONS             *
C*****************************************************************
      X=K
      WRITE(12,847)K
  847 FORMAT(7X,'PERIOD OF SIMULATION = ',I5)
      J=1
      DO 5990 I=X,SMOS,-1
      PDM(J)=P(I,J)
      PDYLD(J)=YN(I,J)
      DO 5970 KK=1,NDIST
      PDY(J,KK)=0.
      PD(J,KK)=0.
      XLD(J,KK)=0.
      IF(KK.LE.15) GOTO 5968
      IF(KK.EQ.16) GOTO 5967
      IF(KK.GT.16) GOTO 5966
 5968 XLD(J,KK)=SLT(J)-((16-KK)*0.2)*SE(J)*NAMP
      PD(J,KK)=PDM(J)*FRQ(KK)
      PDY(J,KK)=PDYLD(J)*FRQ(KK)
      GOTO 5970
 5967 XLD(J,KK)=SLT(J)
      PD(J,KK)=PDM(J)*FRQ(KK)
      PDY(J,KK)=PDYLD(J)*FRQ(KK)
      GOTO 5970
 5966 XLD(J,KK)=SLT(J)+((KK-16)*0.2)*SE(J)*NAMP
      PD(J,KK)=PDM(J)*FRQ(KK)
      PDY(J,KK)=PDYLD(J)*FRQ(KK)
 5970 CONTINUE
 5980 CONTINUE
      J=J+1
```

```
 5990 CONTINUE
      X=K
      J=1
C*****PLOTTING LENGTH @ AGE DISTRIBUTIONS
C         DO 5835 I=X,SMUS,-1
C         DO 5834 KK=1,NDIST
C         XX=J
C         WRITE(7,5833)XX,XLD(J,KK),K
C         WRITE(8,5833)XLD(J,KK),PD(J,KK),K
C         WRITE(12,5833)XLD(J,KK),PDY(J,KK),K
C 5833 FORMAT(F15.7,F15.7,I5)
C 5834 CONTINUE
C         J=J+1
C 5835 CONTINUE
C*************************************************************
C        SEARCHING ROUTINE FOR SCANNING PROBABILISTIC        *
C        LENGTH DISTRIBUTIONS                                *
C*************************************************************
        TSY=0.
        TSLC=0.
        SALC=0.
        XLMAX=0.
        XLMIN=0.
        XMINLEN=0.
        XMAXLEN=0.
        NTOTLI=0.
        RANGEL=0.
        XLMAX=SLT(NM)+50.0*SE(NM)
        XLMIN=SLT(MFR)-12.0*SE(MFR)
        XLMIN=INT(XLMIN)
        XLMAX=INT(XLMAX)
        IF(XLMIN.GE.0) GOTO 5958
        XLMIN=0.0
 5958 NTOTLI=INT((XLMAX-XLMIN)/DTL)
        RANGEL=XLMAX-XLMIN
        DO 5955 ZZ=1,NTOTLI
        XLENPT(ZZ)=0.
        POPLEN(ZZ)=0.
        SYLND(ZZ)=0.
        TSL(ZZ)=0.
        IF(ZZ.NE.1) GOTO 5940
        XMINLEN=XLMIN
        XMAXLEN=XLMIN+DTL
        GOTO 5942
 5940 XMINLEN=XMAXLEN
        XMAXLEN=XMINLEN+DTL
 5942 XLENPT(ZZ)=(XMAXLEN+XMINLEN)/2.
        X=K
        J=1
        DO 5950 I=X,SMOS,-1
        DO 5949 KK=1,NDIST
        IF(J.LT.MFR) GOTO 5949
        IF(XLD(J,KK).LE.XMAXLEN) GOTO 5947
        GOTO 5949
 5947 IF(XLD(J,KK).GE.XMINLEN) GOTO 5945
        GOTO 5949
 5945 POPLEN(ZZ)=POPLEN(ZZ)+PD(J,KK)
        SYLND(ZZ)=SYLND(ZZ)+PDY(J,KK)
        TSL(ZZ)=TSL(ZZ)+(PDY(J,KK)*XLENPT(ZZ))
 5949 CONTINUE
        J=J+1
 5950 CONTINUE
        TSLC=TSLC+TSL(ZZ)
        TSY=TSY+SYLND(ZZ)
 5955 CONTINUE
        SALC=TSLC/TSY
C*************************************************************
C       SEARCHING ROUTINE FOR SCANNING DETERMINISTIC         *
C       POPULATION LENGTH DISTRIBUTIONS                      *
C*************************************************************
        XLMAX=0.
        XLMIN=0.
        NTOTLI=0
        RANGEL=0.
        XMINLEN=0.
        XMAXLEN=0.
        TYP=0.
        TPD=0.
        TDLC=0.
```

```
      DALC=0.
      TYD=0.
      XLMAX=SLT(NM)+50.0*SE(NM)
      XLMIN=SLT(MFR)-12.0*SE(MFR)
      XLMIN=INT(XLMIN)
      XLMAX=INT(XLMAX)
      IF(XLMIN.GE.0) GOTO 6958
      XLMIN=0.
 6958 NTOTLI=INT((XLMAX-XLMIN)/DTL)
      RANGEL=XLMAX-XLMIN
      DO 6955 ZZ=1,NTOTLI
      DPOP(ZZ)=0.
      DYPD(ZZ)=0.
      TDL(ZZ)=0.
      IF(ZZ.NE.1) GOTO 6940
      XMINLEN=XLMIN
      XMAXLEN=XLMIN+DTL
      GOTO 6942
 6940 XMINLEN=XMAXLEN
      XMAXLEN=XMINLEN+DTL
 6942 XDLENPT(ZZ)=(XMAXLEN+XMINLEN)/2.
      X=K
      J=1
      DO 6950 I=X,SMOS,-1
      IF(J.LT.MFR) GOTO 6949
      IF(SLT(J).LE.XMAXLEN) GOTO 6947
      GOTO 6949
 6947 IF(SLT(J).GE.XMINLEN) GOTO 6945
      GOTO 6949
 6945 DPOP(ZZ)=DPOP(ZZ)+P(I,J)
      DYPD(ZZ)=DYPD(ZZ)+YN(I,J)
      TDL(ZZ)=TDL(ZZ)+(YN(I,J)*XDLENPT(ZZ))
 6949 J=J+1
 6950 CONTINUE
      TYD=TYD+DYPD(ZZ)
      TPD=TPD+DPOP(ZZ)
      TDLC=TDLC+TDL(ZZ)
 6955 CONTINUE
      DALC=TDLC/TYD
C**************************************************************
C        WRITING TO DATA FILES                                *
C**************************************************************
      X=K
      J=1
      DO 5820 I=X,SMOS,-1
C      XXXX=J
      IF(J.LT.MFR) GOTO 5819
 5815 WRITE(3,5811)SLT(J),YN(1,J),K
 5811 FORMAT(F15.5,F15.5,I5)
C      WRITE(2,5813)SLT(J),POPP(J),NPLOT
C 5813 FORMAT(F15.4,F15.5,I5)
 5819 J=J+1
 5820 CONTINUE
C      DO 5830 J=1,NTOTLI
C      WRITE(4,5828)XLENPT(J),POPLEN(J),K
C      WRITE(11,5828)XDLENPT(J),DPOP(J),K
C 5830 CONTINUE
 5828 FORMAT(F15.5,F15.5,I5)
      KKK=K+100
      DO 5899 J=1,NTOTLI
      WRITE(11,5828)XDLENPT(J),DYPD(J),KKK
 5899 CONTINUE
      WRITE(11,848)TDLC,TYD,DALC
      KKKK=K+200
      DO 5901 J=1,NTOTLI
      WRITE(12,5828)XLENPT(J),SYLND(J),KKKK
 5901 CONTINUE
      WRITE(12,849)TSLC,TSY,SALC
C      WRITE(11,5836)TPD,TYD,TSY
C 5836 FORMAT(5X,'TOTAL POPN NUMBERS = ',E18.7,/5X,
C    $       'TOTAL YIELD IN NUMBERS = ',E18.7,/5X,
C    $       'TOTAL STOCHASTIC YIELD IN NUMBERS = ',E18.7)
      GOTO 3901
C************************************************************
C                                                           *
C       TRANSITIONAL OUTPUT MODULE                          *
C                                                           *
C************************************************************
 3901 MR=K-MFR
```

```
      IF(R(MR).EQ.0) GOTO 3902
      YPR=YWJ/R(MR)
      GOTO 3903
 3902 YPR=0.0
 3903 WRITE(5,1590)K
 1590 FORMAT(7X,'MONTH = ',I5/)
      XNY=K-NM
      WRITE(9,730)XNY,ALC(K),NPLOT
      WRITE(10,730)XNY,YWJ,NPLOT
      IF(NOUT.EQ.0) GOTO 3008
      IF(NOUT.EQ.1) GOTO 1591
 1591 WRITE(5,1592)
 1592 FORMAT(1X,'AGE',8X,'POPN NO.',3X,'LENGTH',5X,
     *'WEIGHT',7X,'BIOMASS',3X,'YIELD WT.',1X,'YIELD NO.')
      X=K
      J=1
      DO 1700 I = X,SMOS,-1
      WRITE(5,1640)J,P(I,J),SLT(J),WT(J),BIOM(J),YW(I,J),YN(I,J),F(I,J)
 1640 FORMAT(1X,I3,3X,F13.3,2X,F8.2,2X,F8.3,2X,F13.4,2X,F9.4,2X,F8.4,
     $2X,F5.3)
      J=J+1
 1700 CONTINUE
      WRITE(5,1701)PJ,BMJ,YWJ,YNJ
 1701 FORMAT(/1X,'TOTAL',1X,F13.7,22X,E13.7,1X,E13.7,2X,E13.7)
      WRITE(5,1710)AL(K),ALC(K),AVWT(K)
 1710 FORMAT(7X,'AVG. LENGTH POPN = ',F13.6,2X,
     $'AVG. LENGTH CATCH = ',F13.6,/,7X,'AVG. WT. = ',F13.6)
      WRITE(5,1712)YPR
 1712 FORMAT(7X,'YIELD PER-RECRUIT = ',F15.8//)
      GOTO 3060
 3008 WRITE(5,3010)
 3010 FORMAT(7X,'CONDENSED TRANSITIONAL OUTPUT:',//)
      WRITE(5,3020)PJ,BMJ,SSB
 3020 FORMAT(7X,'POPULATION ABUNDANCE = ',F12.3,/,7X,
     $'POPULATION BIOMASS = ',E14.7,/,7X,'SPAWNING STOCK BIOMASS = ',
     $E14.7)
      WRITE(5,3030)YWJ,YNJ
 3030 FORMAT(7X,'YIELD IN WEIGHT = ',F12.4,/,7X,'YIELD IN NUMBERS = ',
     $F12.4)
      WRITE(5,3040)AL(K),ALC(K)
 3040 FORMAT(7X,'AVG. LENGTH IN TOTAL POPULATION = ',F8.4,/,7X,
     $'AVG. LENGTH IN CATCH = ',F8.4)
      WRITE(5,3050)AVWT(K),AVWT2(K)
 3050 FORMAT(7X,'AVG. WT. IN CATCH = ',F8.4,/,7X
     $,'AVG. WT. IN CATCH(2) = ',F8.4)
      WRITE(5,3055)R(K),YPR
 3055 FORMAT(7X,'RECRUITMENT = ',F10.5,/,
     $7X,'YIELD PER RECRUIT = ',F9.4,//)
 3060 IF(JTRANS.NE.12) GOTO 4000
      IF(JTRANS.EQ.12) GOTO 3101
 3101 WRITE(5,3100)
 3100 FORMAT(7X,'AGE',8X,'POPN. NO.',3X,'YWEIGHT',5X,'YNUMBER')
      X=K
      J=1
      DO 3005 I=X,SMOS,-1
      WRITE(5,3004)J,AP(J),AYW(J),AYN(J),SLT(J)
 3004 FORMAT(7X,I3,2X,F10.2,5X,F12.3,5X,F12.3,5X,F12.3)
      J=J+1
 3005 CONTINUE
      WRITE(5,3110)ABMJ,ASSB,APJ
 3110 FORMAT(//,7X,'BIOMASS = ',E14.7,/,7X,'SPAWNING BIOMASS = ',
     $E14.7,/,7X,'POPULATION SIZE = ',E14.7)
      WRITE(5,3120)ATOTWC,AYNJ
 3120 FORMAT(7X,'YIELD IN WEIGHT = ',E14.7,/,7X,'YIELD IN NO. = ',
     $E14.7)
      IF(AYNJ.LE.0) GOTO 3128
      AALC=ATOTLC/AYNJ
      AAVWT=ATOTWC/AYNJ
      GOTO 3129
 3128 AALC=0.
      AAVWT=0.
      GOTO 3129
 3129 WRITE(5,3130)AALC,AAVWT
 3130 FORMAT(7X,'AVG. LENGTH IN CATCH = ',F8.4,5X,'AVG. WEIGHT IN
     $CATCH = ',F8.4)
      WRITE(5,3140)YPR
 3140 FORMAT(7X,'ANNUAL YIELD PER RECRUIT = ',F9.6,//)
      JTRANS=0.
      GOTO 4000
```

```
1107      4000 CONTINUE
1108      4001 END
```

PROGRAM SECTIONS

	NAME	BYTES	ATTRIBUTES
0	$CODE	10058	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	2395	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	1914052	PIC CON REL LCL NOSHR NOEXE RD WRT QUAD
	TOTAL SPACE ALLOCATED	1926505	

ENTRY POINTS

ADDRESS	TYPE	NAME
0-00000000		COREX

VARIABLES

ADDRESS	TYPE	NAME	ADDRESS	TYPE	NAME	ADDRESS	TYPE	NAME	ADDRESS	TYPE	NAME
**	R*4	A1P	**	R*4	A2P	**	R*4	A3P	2-001D3330	R*4	AALC
2-001D3334	R*4	AAVWT	2-001D32F8	R*4	ABMJ	2-001D32B4	R*4	ACBT	2-001D31E4	R*4	ALPHA
2-001D3208	R*4	ALSLB	**	R*4	AMPL	2-001D3310	R*4	APJ	2-001D32FC	R*4	ASSB
2-001D3304	R*4	ATOTLC	2-001D3308	R*4	ATOTLP	2-001D330C	R*4	ATOTWC	**	R*4	AVGL
2-001D3300	R*4	AYNJ	2-001D31E8	R*4	BETA	**	R*4	BETADS	2-001D3298	R*4	BMJ
2-001D320C	R*4	BSLB	2-001D32AC	R*4	CXT	**	R*4	DALC	2-001D3204	R*4	DTL
2-001D31EC	R*4	EOUTL	**	R*4	EYPR	2-001D3240	R*4	FECUN	2-001D31E0	R*4	EMORT
2-001D3210	I*4	I	**	I*4	IV	**	I*4	J	**	I*4	JJ
2-001D3318	I*4	JJJ	2-001D32F4	I*4	JTRANS	2-001D3234	I*4	K	**	I*4	KK
**	I*4	KKK	**	I*4	KKKK	**	I*4	L	**	I*4	L1
2-001D3274	I*4	LCOUNT	**	I*4	M	2-001D31A0	I*4	MEC	2-001D319C	I*4	MFR
2-001D3194	I*4	MOS	**	I*4	MR	2-001D31A4	I*4	MRM	2-001D31C4	I*4	NAMP
2-001D31B4	I*4	NAVA	2-001D31C0	I*4	NDIST	2-001D3190	I*4	NDT	2-001D31A8	I*4	NFOT
2-001D31B8	I*4	NLL	**	I*4	NM	2-001D3214	I*4	NMC	**	I*4	NMT
2-001D31BC	I*4	NOPT	2-001D31B0	I*4	NOUT	2-001D31C8	I*4	NPLOT	2-001D31AC	I*4	NROT
2-001D3198	I*4	NTL	2-001D32C4	I*4	NTOTL1	2-001D31F8	R*4	PARM1	2-001D31FC	R*4	PARM2
2-001D31F4	R*4	PER	**	R*4	PI	2-001D32A0	R*4	PJ	2-001D32B0	R*4	PJCX
2-001D32A8	R*4	PPA	**	R*4	PANGEL	**	R*4	RHO	2-001D329C	R*4	RPJ
2-001D3270	R*4	S2	**	R*4	S3	**	R*4	SALC	2-001D32EC	R*4	SMOS
2-001D3244	R*4	SMEEC	2-001D32A4	R*4	SSB	2-001D326C	R*4	SUM	2-001D31D8	R*4	TO
2-001D32D8	R*4	TPLC	**	R*4	TIME	2-001D3288	R*4	TOTLC	2-001D3284	R*4	TOTLP
2-001D328C	R*4	TOTWC	2-001D32D4	R*4	TPD	2-001D31F0	R*4	TR	2-001D32E[illegible]	R*4	TSLC
2-001D32B8	R*4	TSY	**	R*4	TYD	**	R*4	TYP	2-001D31CC	R*4	ULTNF
**	R*4	VN	**	R*4	VSUM	2-001D31D0	R*4	WINF	2-001D3238	R*4	X
2-001D31D4	R*4	XAKV	**	R*4	XDT	2-001D323C	R*4	XLAMBDA	**	R*4	XLMAX
2-001D32C0	R*4	XLMIN	2-001D32D0	R*4	XMAXLEN	**	R*4	XMINLEN	**	R*4	XN1F
**	R*4	XN2F	**	R*4	XN3F	2-001D31DC	R*4	XNM	2-001D3200	R*4	XNMF
**	R*4	XNY	**	R*4	XX	**	R*4	XXXX	2-001D3294	R*4	YNJ
2-001D332C	R*4	YPR	2-001D3290	R*4	YWJ	2-001D32C8	R*4	ZZ			

ARRAYS

ADDRESS	TYPE	NAME	BYTES	DIMENSIONS
2-0006D420	R*4	A	160000	(200, 200)
2-001D1E70	R*4	ACBIR	800	(200)
2-00131280	R*4	AL	800	(200)
2-001A7870	R*4	ALC	800	(200)
2-001D2E40	R*4	AM	48	(12)
2-001D24B0	R*4	AP	800	(200)
2-001CFC10	R*4	AVWT	800	(200)
2-001D11F0	R*4	AVWT2	800	(200)
2-001D2AF0	R*4	AYN	800	(200)
2-001D27D0	R*4	AYW	800	(200)
2-00131BE0	R*4	BTOM	800	(200)
2-001D1B50	R*4	BTRTHS	800	(200)
2-001D1830	R*4	BTT	800	(200)
2-001D0BB0	R*4	CX	800	(200)
2-00020800	R*8	DPOP	16000	(2000)
2-00024680	R*8	DYPD	16000	(2000)

43

2-00094520	R*4	F	160000	(200, 200)
2-00161210	R*4	FINF	800	(200)
2-001CFF30	R*4	FEC	800	(200)
2-00160EF0	R*4	FM	800	(200)
2-001D2810	R*4	FP	48	(12)
2-00000000	R*8	FRQ	320	(40)
2-0008B620	R*4	P	480000	(600, 200)
2-001D0250	R*4	PBIRTH	800	(200)
2-000007B0	R*8	PD	44800	(160, 35)
2-000201C0	R*8	PDM	1600	(200)
2-0002C380	R*8	PDY	44800	(160, 35)
2-000372B0	R*8	PDYLD	1600	(200)
2-0000B680	R*8	POPLEN	20000	(2500)
2-001D2E70	R*4	POPP	800	(200)
2-00130920	R*4	R	800	(200)
2-001CF5D0	R*4	RA	800	(200)
2-0018BE70	R*4	RBIOM	800	(200)
2-001CF6F0	R*4	RC	800	(200)
2-00159000	R*4	RM	160000	(200, 200)
2-00000140	R*8	SE	1600	(200)
2-00130C40	R*4	SLT	800	(200)
2-0018B1F0	R*4	SLT1	800	(200)
2-00161530	R*4	SXR	800	(200)
2-000378C0	R*8	SYLND	20000	(2500)
2-00041500	R*8	TDL	20000	(2500)
2-001315A0	R*4	TL	800	(200)
2-001A7E90	R*4	TLC	800	(200)
2-0003C6E0	R*8	TSL	20000	(2500)
2-001D06D0	R*4	TWC	800	(200)
2-00130F60	R*4	WT	800	(200)
2-001D0890	R*4	WTF	800	(200)
2-00028500	R*8	XDLENPT	16000	(2000)
2-001CF2B0	R*4	XF	800	(200)
2-001D0570	R*4	XFE	800	(200)
2-000104A0	R*8	XLD	44800	(160, 35)
2-0001B3A0	R*8	XLENPT	20000	(2500)
2-00046320	R*4	XM	160000	(200, 200)
2-001A7200	R*4	XOMEG	16	(4)
2-000FB5A0	R*4	Y	800	(200)
2-001B0100	R*4	YN	160000	(200, 200)
2-00131F00	R*4	YW	160000	(200, 200)
2-001A81B0	R*4	Z	160000	(200, 200)

LABELS

ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL
**	12	**	13	**	15	**	17	0-0000084C	18	0-00000824	19
0-00000872	20	0-00000804	21	1-000000AF	29'	1-00000077	30'	1-000000A6	31'	1-000000AB	33'
**	35	**	40	0-000004F0	41	**	42	**	43	**	45
**	50	**	55	0-0000078C	60	**	61	**	62	**	64
**	67	0-00000B34	68	0-00000B1F	69	0-00000B45	70	**	71	0-00000B4D	79
1-000000B5	80'	0-00000CAC	81	**	82	0-00000D0F	83	0-00000876	84	**	85
0-00000B10	86	0-00000AEC	87	0-00000A18	88	0-00000CBA	89	0-0000089A	90	0-00000460	91
**	92	0-000004BC	95	0-000004E2	96	0-00000490	97	0-000004B7	98	0-000004E5	106
**	107	0-00000DC1	108	0-00000E35	109	0-00000C10	110	0-00000D08	111	0-00000D6D	112
**	120	0-000006F3	130	0-00000754	131	0-00000724	135	0-0000077A	141	0-0000074E	142
0-0000077D	148	**	149	0-00000E58	150	**	184	**	186	**	187
**	188	**	189	0-00000562	200	**	201	0-00000640	202	0-00000614	203
0-0000067F	205	**	206	**	207	**	216	**	217	**	218
**	219	**	220	**	430	**	431	**	500	1-0000020D	590'
1-0000022F	591'	1-0000027C	600'	1-0000029E	610'	1-000002EB	640'	1-00000313	650'	**	700
1-0000033B	701'	1-0000035A	702'	1-00000379	710'	1-000003C2	711'	1-00000423	712'	1-000004D7	713'
1-0000051A	714'	1-0000053C	715'	1-0000046E	716'	**	720	1-00000533	730'	0-00000EE7	798
**	799	0-00000F3C	802	0-00000F45	803	**	810	1-000000E6	811'	**	813'
**	815	0-00001398	819	**	820	1-000005A8	821'	0-00001CF6	824	**	825
1-000000EF	828'	**	830	**	833'	**	834	**	835	1-000001A3	836'
0-00001487	840	1-00000591	847'	1-000000F8	848'	1-0000014F	849'	**	899	**	901
0-00001150	940	0-0000115B	942	0-000011B8	945	0-000011A8	947	0-000011DB	949	**	950
**	955	0-00001100	958	0-00000FAA	966	0-00000FFC	967	0-0000102C	968	0-0000107B	970
**	980	**	990	0-00000EA4	1000	**	1430	**	1431	0-000018DB	1501
1-00000554	1502'	0-00001BE9	1509	0-00001B3C	1511	0-00001B43	1512	0-00001AF9	1515	0-00001C17	1550
0-00001A8C	1581	**	1582	0-00001A9A	1583	1-000005C1	1590'	0-00002335	1591	1-000005D1	1592'
**	1600	0-000019F0	1610	**	1620	1-0000061E	1640'	**	1700	1-00000646	1701'
1-00000665	1710'	1-000006AE	1712'	0-00001274	1940	0-0000127B	1942	0-000012B4	1945	0-000012A8	1947
0-000012FA	1949	**	1950	**	1955	0-00001228	1958	1-00000866	3004'	**	3005
0-000024B4	3008	1-000006CC	3010'	1-000006F1	3020'	1-0000074E	3030'	1-000007B3	3040'	1-000007CC	3050'
1-00000807	3055'	0-000025D5	3060	1-0000083B	3100'	0-000025E1	3101	1-0000087F	3110'	1-000008C8	3120'

```
0-000026F4  3128    0-000026EC  3129    1-000008F9  3130'   1-00000936  3140'   0-00001C67  3900            **  3901
0-000022B8  3902    0-000022BC  3903    0-0000198D  3998            **  3999    0-0000273D  4000    0-00002746  4001
1-000005AF  5811'           **  5815    0-00002146  5819            **  5820    1-000005B8  5828'           **  5899
        **  5901    0-00001ECC  5940    0-00001ED3  5942    0-00001F2C  5945    0-00001F1C  5947    0-00001F4F  5949
        **  5950            **  5955    0-00001E7A  5958    0-00001D7A  5966    0-00001DBC  5967    0-00001DD8  5968
0-00001E16  5970            **  5980            **  5990    0-00002010  6940    0-00002017  6942    0-00002054  6945
0-00002048  6947    0-0000209A  6949            **  6950            **  6955    0-00001FB9  6958


FUNCTIONS AND SUBROUTINES REFERENCED

  TYPE  NAME                TYPE  NAME              TYPE  NAME              TYPE  NAME

        FOR$OPEN            R*4   MTH$EXP           R*4   MTH$SIN           R*4   MTH$SQRT


COMMAND QUALIFIERS

  FOR/LIS COREX

  /CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /STANDARD=(NOSYNTAX,NOSOURCE_FORM)
  /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
  /WARNINGS=(GENERAL,NODECLARATIONS,NOULTRIX)
  /CONTINUATIONS=19  /NOCROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE  /NOANALYSIS


COMPILATION STATISTICS

  RUN TIME:             25.16 SECONDS
  ELAPSED TIME:         27.86 SECONDS
  PAGE FAULTS:          8416
  DYNAMIC MEMORY:       1354 PAGES
```

GROIO.DAS

12	0	13	1	13	24	2	4
0	1	14	1	31	8	2	2.279044
1290.0	25.032719	.122	-1.127	0.2	(0.1)	26871.102	.10398
1.	12000.	12.	4.5	4.5	6.0	1.0	1.0
7.5865E-01	0.	0.	1.0	1.0	1.0	1.0	1.
0.	0.	0.	0.	1.	1.	1.	.0222
1.0	1.	1.	1.	.0072	.0108	.0159	.0796
1.	1.	1.	1.	.0665	.0736	.0781	.0222
1.	.0016	.0027	.0045	.0484	.0389	.03	
.0009	.0389	.0484	.0579	.0027	.0016	.0009	
.03	.0736	.0665	.0579				
.0781	.0108	.0072	.0045				
.0159							

$L_\infty = 1290$

$t_0 = -1.127$

$K = .122$

$K/12 =$

LINF,rar

$L_c = 299.88589478$

$L_\lambda = 1058.63720703$

$K/12 \equiv .01016667$

$z/12 \equiv .025$

46

SCROID.DAT

12	0	10	11	11	13	2	2
1	1	14	1	31	8	7	
1690.0	99.1272728	.600 0.148	.8333333	0.8	(1.6) F	26871.102	2.279044
0.	12000.	12.	4.5	4.5	6.0	40.0	.10398
7.5865E-01							
0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0
1.0	0.0	0.0	0.0				
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0				
.0009	.0016	.0027	.0045	.0072	.0108	.0159	.0222
.03	.0389	.0484	.0579	.0665	.0736	.0781	.0796
.0781	.0736	.0665	.0579	.0484	.0389	.03	.0222
.0159	.0108	.0072	.0045	.0027	.0016	.0009	

4

```
          12         11           6          24         24        24          2         4
           1          1           7           1         31         8          7         1
1250.0     17.076065 .400        -.1443000 0.635       0.56       26871.102 2.279044
0.         12000.     24.         4.5        4.5        6.0       20.0       .10398
.7587           .00000000900      3.150
0.0        0.0        0.0         1.0        1.0        1.0       1.0        1.0
1.0        0.0        0.0         0.0
1.0        1.0        1.0         1.0        1.0        1.0       1.0        1.0
1.0        1.0        1.0         1.0
.0009      .0016      .0027       .0045      .0072      .0108     .0159      .0222
.03        .0389      .0484       .0579      .0665      .0736     .0781      .0796
.0781      .0736      .0665       .0579      .0484      .0389     .03        .0222
.0159      .0108      .0072       .0045      .0027      .0016     .0009
```

48

CORVINA.DAT

·TUNA = SCROID.DAT

```
           12          0          10          11          11         13           2
            1          1          14           1          31          8           7
1690.0     99.127273 .600        .83333333 0.8         0.1         26871.102 2.279044
0.         12000.     12.        4.5         4.5         6.0         40.0        .10398
.7587               .00000003894 3.020
0.0        0.0        0.0        1.0         1.0         1.0         1.0         1.0
1.0        0.0        0.0        0.0
1.0        1.0        1.0        1.0         1.0         1.0         1.0         1.0
1.0        1.0        1.0        1.0
.0009      .0016      .0027      .0045       .0072       .0108       .0159       .0222
.03        .0389      .0484      .0579       .0665       .0736       .0781       .0796
.0781      .0736      .0665      .0579       .0484       .0389       .03         .0222
.0159      .0108      .0072      .0045       .0027       .0016       .0009
```

Jj*6Bbb

G&

GROID.DAT

8-7-87

	12	12	13	3	13	24	2
	0	1	14	1	31	8	7
1290.0	25.032719	.122	-1.127	0.2	0.1	10899.525	2.5807
0.	12000.	24.	4.5	4.5	6.0	40.0	.10398
.7587		.00000001200	2.996				
0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0
1.0	0.0	0.0	0.0				
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0				
.0009	.0016	.0027	.0045	.0072	.0108	.0159	.0222
.03	.0389	.0484	.0579	.0665	.0736	.0781	.0796
.0781	.0736	.0665	.0579	.0484	.0389	.03	.0222
.0159	.0108	.0072	.0045	.0027	.0016	.0009	