Metapopulation Assessment System Case Study 1 Vignette

Jon Brodziak and Matthew Supernaw

NOAA Fisheries, PIFSC

1845 Wasp Blvd., Honolulu, HI

Jon.Brodziak@NOAA.GOV and Matthew.Supernaw@NOAA.GOV

16-March-2018

The goal of case study 1 of the metapopulation assessment system (MAS) is to implement and test the data types, model structures and parameters needed to initialize and simulate a dynamic metapopulation model. That is, create the structures and test the algorithms needed to construct an operating model of a simple metapopulation fishery system. This includes an algorithm to calculate unfished equilibrium conditions that account for the spatial distribution of recruitment and the movement of fish among areas, an algorithm to calculate fished equilibrium conditions to initialize the metapopulation and an algorithm to calculate dynamic fished conditions throughout the assessment time horizon.

To do this, we developed a 2-area by 2-population by 2-fleet metapopulation fishery system for case study 1. This model represents a dynamic spatial fishery system with a north-south latitudinal cline for the population characteristics between two areas. The two populations are from a hypothetical migratory groundfish species with 9 true age classes comprised of ages 0 to 8 years and a plus group for ages 9 and older (the plus group index is *A*=9+). The time horizon for assessment is 20 years with an annual time step and a single season is used for fishery dynamics modeling. Both populations are modeled as having two sexes but there is no difference between genders in either population.

The two populations inhabit two areas and have distinct recruitment and demographic trajectories. Area A is more temperate and area B is more southerly and subtropical. Both populations spawn in both areas A and B. The two populations engage in feeding migrations between areas and both populations have life history characteristics that persist when individuals move between areas.

A mixed-population fishery exists in both areas with one fishing fleet in area A and one fleet in area B. Both fleets have time-invariant operating characteristics during the assessment time horizon and fishery selectivity at age is time-invariant for each fleet (Table 1). The fishing mortality rate exerted by both fleets changes through four time periods; the initial unfished equilibrium period, the initial fished equilibrium period, the first 10 years of the assessment time period, and the second 10 year of the assessment time period.

Table 1. Fishing mortality rates by fleet/area and time period.

	Unfished Equilibrium Period	Fished Equilibrium Period	Assessment Period 1	Assessment Period 2
Duration of Period	Iterations to Convergence =48	Iterations to Convergence=88	Year=1 to 10	Year=11 to 20
Fleet A Fishing Mortality Rate	$F_{unfished,Area\ A} = 0$	$F_{fished,Area\ A} = 0.1$	$F_{Period\ 1,Area\ A} = 0.4$	$F_{Period\ 2,Area\ A} = 0.2$
Fleet B Fishing Mortality Rate	$F_{unfished,Area\ B} = 0$	$F_{fished,Area\ B} = 0.1$	$F_{Period\ 1,Area\ B} = 0.2$	$F_{Period\ 2,Area\ B} = 0.2$

A fishery-independent survey is conducted in both areas to measure the combined relative abundances of the two populations. The survey has consistent operating characteristics and areal coverage throughout the assessment time horizon and the survey selectivity at age for each population is time-invariant.

The simulation algorithm proceeds by reading in the input and output file names from the console. The next step is to read the input data from the input file and write the input data to the output file. The next step is to calculate the unfished and fished equilibrium conditions and write the results to the output file. The next step is to calculate the system dynamics through the assessment period. The last step is to append the results of the assessment period simulations to the output file.

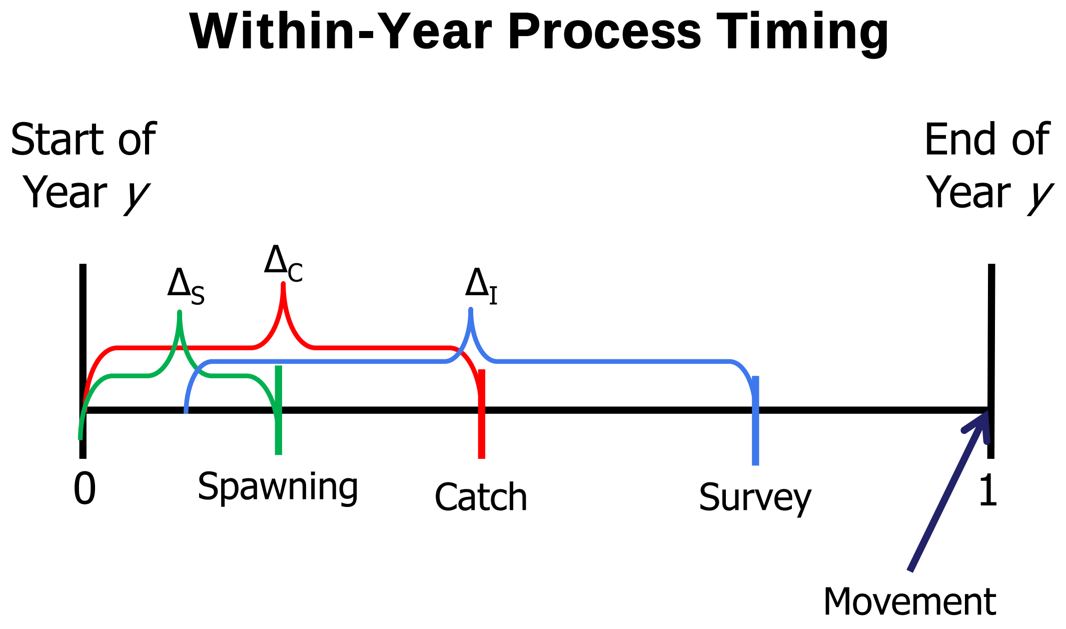
1. Population Dynamics Equations

The two populations are modeled as numbers of individuals by age and gender for each time period with an annual time step. Population-level numbers are reported in units of 1000s. The biomass of individuals has units of kg while population-level values have units of metric tons, noting that all units are SI. Here are the details of the processes needed to set up the two-population model.

1.1 Within-Year Process Timing

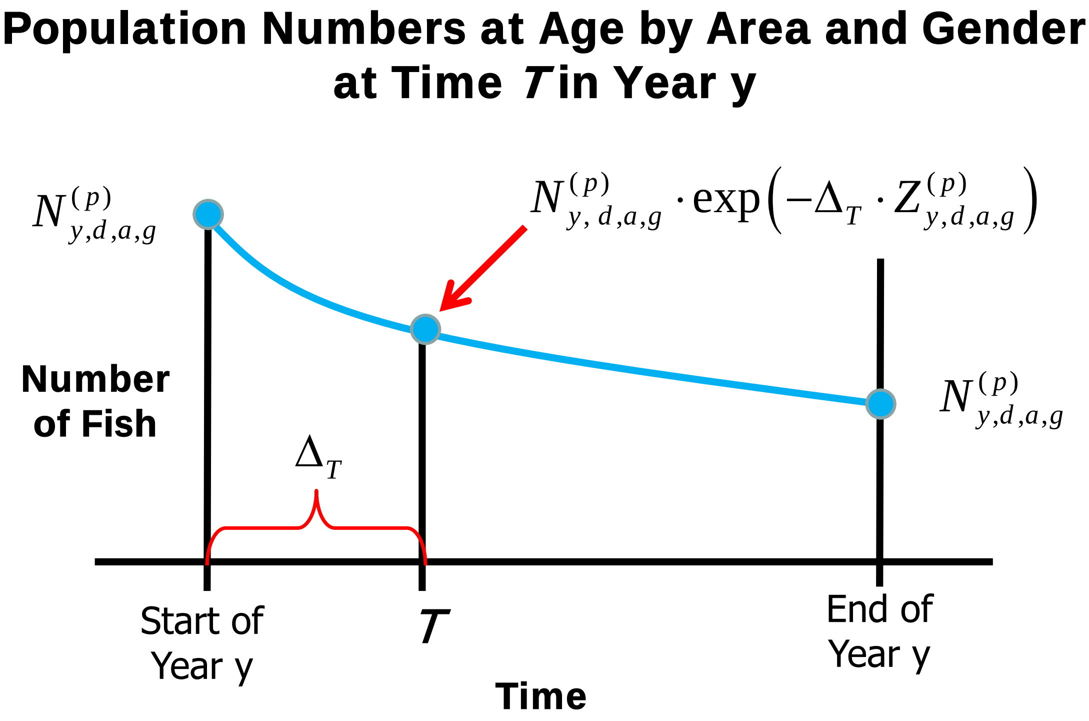
The timings of population processes for the processes of spawning, fishery catch, and surveying by area within a year are identical for populations 1 and 2. The timing of the spawning process occurs at the end of the first 3 months, or one quarter of the year, with a within-year recruitment-spawning offset of $\Delta_S = 0.25$ from the start of each year. The timing of the catch process is set to occur at the mid-point of the year with a within-year catch offset of $\Delta_C = 0.5$. The timing of the population survey process occurs at the end of the third quarter of the year with a within-year survey offset of $\Delta_I = 0.75$. Population movement occurs at the end of the year.

Figure 1. The within-year timing for the processes of spawning, fishery catch, and surveying.



Given the within-year timing of the processes of spawning, catch, growth, survey and movement, the numbers at age by population, area, and gender are calculated based on the exponential decrease in numbers at age under the total mortality rate and process offset within the year (Figure 2).

Figure 2. The seasonal survival process for population numbers at age by year, area, and gender.


Population Numbers at Age by Area and Gender
at Time *T* in Year y
$N^{(p)}_{y,d,a,g}$
$N^{(p)}_{y,\,d,a,g}\cdot\exp\left(-\Delta_T\cdot Z^{(p)}_{y,d,a,g}\right)$
$N^{(p)}_{y,d,a,g}$
Number of Fish
Δ_T
Start of Year y
T
End of Year y
Time

Life history processes for growth, weight at length, natural mortality, and maturation, are population-specific, with the exception of the recruitment process which has area-specific parameters for each population. We use the convention that indexes for population and fishery characteristics are labeled as follows: *p* is population, *d* is area, *y* is year, *a* is age, *b* is age bin, *g* is gender, *v* is fleet and *I* is survey.

1.2 Growth Process

Growth in the populations is modeled using a modified von Bertalanffy growth curve for mean length at age *a* by population and gender at time Δ within a year $\left(L^{(p)}_g(a+\Delta)\right)$ where $a_{\min}=1$ *and* $a_{\max}=7$ years of age and female and male growth parameters are equal in each population (Table 2). For the true age classes with ages less than the plus group age, *a*<*A*, the mean length at age $a+\Delta$ (e.g., Figure 1) for gender *g* in population *p* is

$$L_g^{(p)}\left(a+\Delta\right) \;=\; L_{\min,g}^{(p)} + \left(L_{\max,g}^{(p)} - L_{\min,g}^{(p)}\right)\cdot \frac{\left(1-\left(c_g^{(p)}\right)^{a+\Delta-a_{\min}}\right)}{\left(1-\left(c_g^{(p)}\right)^{a_{\max}-a_{\min}}\right)}$$

Table 2. Growth parameters by population.

	Length at $a_{\min}$, $L_{\min}^{(p)}$	Length at $a_{\max}$, $L_{\max}^{(p)}$	Curvature Coefficient $c^{(p)}$
Population 1	20 cm	70 cm	0.5
Population 2	20 cm	65 cm	0.5

1.3 Equilibrium Mean Length of the Plus Group Algorithm

For the plus group which is comprised of all fish with ages greater than or equal to a cutoff age denoted as age-*A*, we want to account for any expected growth in length of plus group fish that occurs beyond age-*A* at equilibrium with total mortality rates at age Z_a. To do this for the equilibrium case, let N^* be the total number of fish at an equilibrium total mortality rate Z_A in the plus group of population *p* in year *y*. Here the plus group consists of ages *A* (the reference or youngest age in the plus group) to *MaxPlusGroupAge* (the oldest possible age of a fish in the plus group) with *MaxPlusGroupAge* >= *A* where the total number of fish in the plus group is the sum of numbers at age *A* to *MaxPlusGroupAge*, $N^* = \sum_{k=A}^{MaxPlusGroupAge} N_k$. For this case study, we set *MaxPlusGroupAge*=20 years to illustrate this population model attribute.

Now we want to approximate the expected length of the plus group when the population is at equilibrium, denoted as $E\left[L_A^*\right] \equiv E\left[L_{A,t=0}^*\right]$ using an abundance-weighted average of the mean lengths of survivors in the plus group, where the mean length of an age-*a* fish is L_a. To do this, assume that (i) all age classes in the plus group have equal abundance at age-*A* when they entered the plus group (N_A) and (ii) all age classes in the plus group experience the same equilibrium instantaneous total mortality rate, denoted by Z_A, where the number of fish at age *A*+*k*+1 is the number of survivors from age *A*+*k*, or $N_{A+k+1} = N_{A+k}\cdot e^{-Z_A}$.

The total number of fish in the plus group can be expressed as the sum of survivors at age for ages *A* to *MaxPlusGroupAge* as

$$N^* \;=\; N_A + N_{A+1} + N_{A+2} + \ldots + N_{MaxPlusGroupAge} \;=\; N_A\cdot\left(1+e^{-Z_A}+e^{-2Z_A}+\ldots+e^{-(MaxPlusGroupAge-A-1)\cdot Z_A}\right)$$

Summing the finite geometric series of *MaxPlusGroupAge*+1 terms on the RHS above gives the following expression for the total number of fish in the plus group

$$N^{*} \;=\; N_{A}\cdot\frac{\left(1-e^{-(MaxPlusGroupAge-A)\cdot Z_{A}}\right)}{\left(1-e^{-Z_{A}}\right)}$$

As a result, the expected length of the set of fish in the plus group ($E\left[L_{A}^{*}\right]$) at equilibrium total mortality rate Z_{A} can be expressed as a weighted average of mean lengths of survivors at age as

$$E\left[L_{A}^{*}\right]=\frac{\sum_{k=A}^{MaxPlusAge} N_{k}\cdot L_{k}}{N^{*}}=\frac{\left(1-e^{-Z_{A}}\right)}{\left(1-e^{-(MaxPlusGroupAge-A)\cdot Z_{A}}\right)}\cdot\sum_{k=0}^{MaxPlusGroupAge-A-1} e^{-k\cdot Z_{A}}\cdot L_{A+k}$$

1.4 Time-Varying Mean Length of the Plus Group Algorithm

As year class strength and survival rates vary across cohorts in the population, the number of new fish entering the plus group will change dynamically. One can improve the accuracy of the plus group dynamics by accounting for the dynamic effects of varying year class strengths and survival rates on the mean length of the plus group through time. To do this, one can approximate the time-varying expected length of the plus group using a weighted average of the mean lengths of the age-(*A*-1) age class entering the plus group at ($N_{A-1,t}$) and the total survivors in the plus group ($N_{A,t}^{*}$) where the (possibly) time-varying mean length of an age-*A* fish at time *t* is $L_{A,t}$ and the expected length of the entire plus group at time *t* is $E\left[L_{A,t}^{*}\right]$. Given this, the expected length of the plus group at time *t*+1 can be approximated as a weighted average of the mean length of age-(*A*-1) survivors entering the plus group at time *t*+1 and the expected length of the plus group survivors from the previous time step $E\left[L_{A,t}^{*}\right]$ as

$$E\left[L_{A,t+1}^{*}\right] \;=\; w_{A,t}\cdot L_{A,t+1}+w_{A,t}^{*}\cdot E\left[L_{A,t+1}^{*}\right]=\frac{N_{A-1,t}\cdot e^{-Z_{A-1,t}}\cdot L_{A,t+1}+N_{A,t}^{*}\cdot e^{-Z_{A,t}}\cdot E\left[L_{A,t}^{*}\right]}{N_{A-1,t}\cdot e^{-Z_{A-1,t}}+N_{A,t}^{*}\cdot e^{-Z_{A,t}}}$$

Here the dynamic weights $w_{A,t}$ are the fraction of the incoming age-*A* fish and the complementary fraction of existing age-(*A*+1) and older fish alive at time *t*+1 in the extended plus group.

For example, the expected length of the plus group at equilibrium with total mortality rate $Z_{A,0}$ at time *t*=0 is

$$E\left[L^*_{A,0}\right] = \frac{\left(1-e^{-Z_{A,0}}\right)}{\left(1-e^{-(MaxPlusAge-A)\cdot Z_{A,0}}\right)} \cdot \sum_{k=0}^{MaxPlusAge-A-1} e^{-k\cdot Z_{A,0}} \cdot L_{A+k,0}$$

which is set to be equal to the plus group mean length at time *t*=1 to start the model, or $L_{A,1} = E\left[L^*_{A,0}\right]$.

Given this initial condition, the expression for the expected mean length of the plus group in the initial time period of the assessment time horizon is

$$E\left[L^*_{A,1}\right] = w_{A,1} \cdot L_{A,1} + w^*_{A,1} \cdot E\left[L^*_{A,0}\right] = E\left[L^*_{A,0}\right]$$

While the corresponding dynamic mean lengths at times T=2…*Y* are

$$E\left[L^*_{A,T}\right] = w_{A,T} \cdot L_{A,T} + w^*_{A,T} \cdot E\left[L^*_{A,T-1}\right] = \frac{N_{A-1,T-1} \cdot e^{-Z_{A-1,T-1}} \cdot L_{A,T} + N^*_{A,T-1} \cdot e^{-Z_{A,T-1}} \cdot E\left[L^*_{A,T-1}\right]}{N_{A-1,T-1} \cdot e^{-Z_{A-1,T-1}} + N^*_{A,T-1} \cdot e^{-Z_{A,T-1}}}$$

And so on for *t*>2 throughout the assessment time horizon.

Note that this time series of calculated dynamic mean lengths of the plus group accounts for the differences in survival and growth for the incoming age-(*A*-1) cohort to the plus group but not potential changes in mean size at age within the extended plus group due to time-varying growth.

1.5 Length-Weight Process

Weight-at-length in the populations is modeled as an allometric relationship where individual fish weights (kg) are a function of length at age (cm), denoted as $W\left(L(a)\right)$, and where female and male length-weight parameters ($A^{(p)}$ and $B^{(p)}$) are equal in each population (Table 3).

For true age classes *a*<*A*, the mean weight at the length *L* at age-*a* in population *p* is

$$W^{(p)}\left(L\right) = A^{(p)} \cdot L^{B^{(p)}}$$

Given the approximation of the expected length of the plus group in population *p*, the expected weight of the plus group $E\left[W^{(p)}\left(L\left(A\right)\right)\right]$ as a function of length at age for the extended plus group is approximated as

$$E\left[W^{(p)}\left(L(A)\right)\right] = \frac{\left(1-e^{-Z_A}\right)}{\left(1-e^{-MaxAge\cdot Z_A}\right)} \sum_{k=0}^{MaxAge-1} e^{-k\cdot Z_A} \cdot A^{(p)} \cdot L\left(A+k\right)^{B^{(p)}}$$

Table 3. Length-weight parameters by population.

	$A^{(p)}$	$B^{(p)}$
Population 1	$2.5\cdot 10^{-5}$	3.0
Population 2	$2.5\cdot 10^{-5}$	2.95

1.6 Natural Mortality Process

Natural mortality rates by age and gender in the populations ($M_{a,g}^{(p)}$) are modeled as constant values through time and area where female and male natural mortality parameters are equal in each population (Table 4).

Table 4. Natural mortality rates at age by population.

	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_{9+}
Population 1	0.5	0.3	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
Population 2	0.55	0.35	0.25	0.2	0.2	0.2	0.2	0.2	0.2	0.2

1.7 Maturation Process

Fraction mature at age and gender in the populations ($P_{mature,g}^{(p)}(a)$) is modeled as a logistic function of age where female and male maturity parameters are equal in each population through time and space (Table 5).

$$P_{mature,g}^{(p)}(a) = \frac{1}{1+\exp\left(-\frac{\left(a-a_{50,g}^{(p)}\right)}{\sigma_{mature,g}^{(p)}}\right)}$$

Table 5. Logistic maturity ogive parameters by population.

	Age at 50% Maturity Parameter a_{50}	Slope Parameter σ_{mature}
Population 1	3.0	0.1
Population 2	3.0	0.5

1.8 Recruitment Production Process

The expected value of the recruitment process in the populations is modeled with a Beverton-Holt stock-recruitment function by population, area, and year. Recruitment produced by population in year *y* and area *d* $\left(R_{y,d}^{(p)}\right)$ is the total number of female and male offspring spawned in area *d* using the area-specific stock-recruitment parameters (Table 6) for the steepness form of the Beverton-Holt curve

$$R_{y,d}^{(p)} = \frac{4h_d^{(p)} \cdot R_{unfished,d}^{(p)} \cdot SB_{y,d}^{(p)}}{SB_{unfished,d}^{(p)}\left(1-h_d^{(p)}\right)+SB_{y,d}^{(p)}\left(5h_d^{(p)}-1\right)} = f_d^{(p)}\left(SB_{y,d}^{(p)} \,|SB_{unfished,d}^{(p)}, R_{unfished,d}^{(p)}, h_d^{(p)}\right)$$

Table 6. Recruitment parameters by population and area including *MaxPlusGroupAge* adjustment with *MaxPlusGroupAge*=20 years.

	Spawning Area	Unfished Equilibrium Spawning Biomass (mt) $SB_{unfished,d}^{(p)}$	Fished Equilibrium Spawning Biomass (mt) $SB_{fished,d}^{(p)}$	Unfished Recruitment (thousands of age-0 fish) $R_{unfished,d}^{(p)}$	Stock-recruitment steepness parameter $h_d^{(p)}$
Population 1	Area A	55878	35429	10000	0.75
	Area B	49528	30523	5000	0.75
Population 2	Area A	39128	24431	10000	0.7
	Area B	39128	24458	10000	0.8

1.9 Spatial Distribution of Recruitment Process

The spatial distribution of the recruitment produced by each population is a function of the constant population-specific recruitment distribution matrices $\underline{\underline{Q}}^{(p)} = \left(Q_{i \to j}^{(p)}\right)$ where $Q_{i \to j}^{(p)}$ is the probability that recruits produced by population *p* in area *i* that are distributed to area *j* (Table 7) and the gender fraction $\rho_g^{(p)}$ where the number of age-0 recruits in year *y* by area and gender $R_{y,d,g}^{(p)}$ is

$$R_{y,d,g}^{(p)} = \rho_g^{(p)} \left(Q_{d \to d}^{(p)} R_{y,d}^{(p)} + \sum_{k \neq d} Q_{k \to d}^{(p)} R_{y,k}^{(p)} \right)$$

Table 7. Recruitment distribution parameters by population.

	Area A		Area B	
	$Q_{A \to A}$	$Q_{A \to B}$	$Q_{B \to B}$	$Q_{B \to A}$
Population 1	0.9	0.1	0.9	0.1
Population 2	0.75	0.25	0.75	0.25

1.10 Population Movement Process

The population movement probabilities are a function of the constant population-specific box transfer matrices $\underline{\underline{T}}^{(p)} = \left(T_{i \to j}^{(p)}\right)$ where $T_{i \to j}^{(p)}$ is the probability that a fish from population *p* in area *i* moves to area *j* (Table 8) in a unit time step.

Table 8. Movement probability parameters by population.

	Area A		Area B	
	$T_{A \to A}$	$T_{A \to B}$	$T_{B \to B}$	$T_{B \to A}$
Population 1	0.9	0.1	0.9	0.1
Population 2	0.75	0.25	0.75	0.25

1.11 Fishery Selectivity Process

The fishery selectivity is approximated with time-invariant logistic fishery selectivity ogives at age by fleet, population, and gender for the proportion of individuals selected at age *a* by fleet *v* $\left(S_{v,g}^{(p)}(a)\right)$ where female and male fishery selectivity parameters are equal (Table 9).

$$S_{v,g}^{(p)}(a)=\frac{1}{1+\exp\left(-\frac{\left(a-a_{50,v,g}^{(p)}\right)}{\sigma_{v,g}^{(p)}}\right)}$$

Table 9. Logistic fishery selectivity at age parameters by fleet.

	Age at 50% Fishery Selectivity Parameter $a_{50,v}$	Fishery Selectivity Slope Parameter σ_v
Fleet A	3.0	0.1
Fleet B	3.0	0.5

1.12 Survey Selectivity Process

The survey selectivity is approximated with time-invariant logistic survey selectivity ogives at age by fleet, population, and gender for the proportion of individuals selected at age *a* by the survey (indexed by *I*) in area $d\left(S_{I,d,g}^{(p)}(a)\right)$ where female and male fishery selectivity parameters are equal (Table 10).

$$S_{I,d,g}^{(p)}(a)=\frac{1}{1+\exp\left(-\frac{\left(a-a_{50,I,d,g}^{(p)}\right)}{\sigma_{I,d,g}^{(p)}}\right)}$$

Table 10. Logistic survey selectivity at age parameters by population.

	Age at 50% Survey Selectivity Parameter $a_{50,I}$	Survey Selectivity Slope Parameter σ_I
Population 1	3.0	0.25
Population 2	2.75	0.5

1.13 Total Mortality Process

The total instantaneous mortality rate at age *a* process by year, population, area, and gender $\left(Z_{y,d,a,g}^{(p)}\right)$ is modeled as the sum of natural mortality and fishing mortality at age *a* by population, year, area (same as fleet index), and gender where

$$Z_{y,d,a,g}^{(p)} = M_{a,g}^{(p)} + S_{d,a,g}^{(p)} F_{y,d,g}^{(p)}$$

1.14 Total Survival Process

The total survival at age *a* process for successive years by population, area, and gender is based on the total mortality process. For true age classes, the year-to-year changes in numbers at age *a* by population, year, area, and gender $\left(N_{y,d,a,g}^{(p)}\right)$ depend on the number of survivors at age *a*-1 from the previous year that move to area *d* through the relationship

$$N_{y,d,a,g}^{(p)} = \sum_k N_{y-1,k,a-1,g}^{(p)} \cdot \exp\left(-Z_{y-1,k,a-1,g}^{(p)}\right) \cdot T_{k \to d}^{(p)}$$

For the age-9+ plus group, the number of survivors at age *A*=*9*+ is a function of survivors at age-(*A*-1) and age-*A* from the previous year that move to area *d* via

$$N_{y,d,A,g}^{(p)} = \sum_k N_{y-1,d,A,g}^{(p)} \cdot \exp\left(-Z_{y-1,d,A,g}^{(p)}\right) \cdot T_{k \to d}^{(p)} + \sum_k N_{y-1,d,A-1,g}^{(p)} \cdot \exp\left(-Z_{y-1,d,A-1,g}^{(p)}\right) \cdot T_{k \to d}^{(p)}$$

1.15 Female Spawning Biomass Process

Female spawning biomass by population and year and area $\left(SB_{y,d,g=female}^{(p)}\right)$ is derived from fractions mature, spawning weights, total mortalities, and population numbers at age via

$$SB_{y,d,g=female}^{(p)} = \sum_a P_{mature,a,g=female}^{(p)} \cdot W_{spawn,a,g=female}^{(p)} \cdot N_{y,d,a,g=female}^{(p)} \cdot \exp\left(-\Delta_S Z_{y,d,a,g=female}^{(p)}\right)$$

1.16 Unfished Survival to Age Process

The unfished survival to age probabilities by population and gender $\left(P_{survive,a,g}^{(p)}\right)$ are derived from the natural mortality rates by age and gender via

$$P_{survive,a,g}^{(p)} = \prod_{i=0}^{a-1} \exp\left(-M_{i,g}^{(p)}\right)$$

1.17 Unfished Numbers at Age Process

The unfished numbers at age in equilibrium by population, area, and gender $\left(N_{unfished,d,g}^{(p)}\right)$ are derived from unfished recruitment by population and area $\left(R_{unfished,d}^{(p)}\right)$, the sex ratio by population (ρ_g), and the natural mortality rates by population, age, and gender via

$$N_{unfished,d,a=0,g}^{(p)} = \rho_g^{(p)} \cdot \sum_k R_{unfished,k}^{(p)} \cdot Q_{k\to d}^{(p)}$$

$$N_{unfished,d,a,g}^{(p)} = \sum_k N_{unfished,k,a-1,g}^{(p)} \exp\left(-M_{a-1,g}^{(p)}\right) \cdot T_{k\to d}^{(p)}$$

$$N_{unfished,d,A,g}^{(p)} = \sum_k N_{unfished,k,A-1,g}^{(p)} \exp\left(-M_{A-1,g}^{(p)}\right) \cdot T_{k\to d}^{(p)} + \sum_k N_{unfished,k,A,g}^{(p)} \exp\left(-M_{A,g}^{(p)}\right) \cdot T_{k\to d}^{(p)}$$

Note that the unfished recruitment by population and area depends on the unfished female spawning biomass by population and area.

1.18 Unfished Female Spawning Biomass Process

The amount of unfished female spawning biomass in equilibrium by population, area, and gender $\left(SB_{unfished,d,g}^{(p)}\right)$ as a function of unfished population numbers and natural mortality at age via

$$SB_{unfished,d,g}^{(p)} = \sum_a P_{mature,a,g}^{(p)} \cdot W_{spawn,a,g}^{(p)} N_{unfished,d,a,g}^{(p)} \cdot \exp\left(-\Delta_S M_{a,g}^{(p)}\right)$$

Note that the unfished female spawning biomass by population and area depends on the unfished numbers at age by population and area.

Because unfished spawning biomasses by population and area are derived parameters, one needs to calculate the associated unfished numbers at age by population and area to determine its value for the recruitment modeling process. Now, the unfished numbers at age can be calculated if the recruitment and movement dynamics produce a stable fixed point for an iterated map of the unfished equilibrium population dynamics. In some cases of recruitment and movement dynamics, a stable fixed point will not exist.

2.1 Unfished Equilibrium Numbers at Age Algorithm

In what follows, we provide details of an algorithm to iteratively calculate unfished equilibrium numbers at age, conditioned on the existence of an equilibrium solution. That is, we need to calculate unfished numbers at age in equilibrium by population, area, and gender $\left(N_{unfished,d,a,g}^{(p)}\right)$ to determine the values of unfished female spawning biomasses by population and area to inform the recruitment process models. Here note that the unfished numbers at age by population, area, and gender depend on the population movement and recruitment distribution matrices and are needed to compute the unfished spawning biomasses by population and area, which in turn, are needed to implement the recruitment submodels by population and area for the initial fished equilibrium time period and assessment time horizon. That is, this algorithm will determine the values of the unfished equilibrium female spawning biomasses for the recruitment submodels by population and area, which are derived quantities that depend on the unfished recruitment parameters by population and area.

Iteration i=1: Calculate the initial unfished numbers at age estimates by population, area, and gender based on unfished recruitment, the recruitment distribution, equilibrium natural mortality and no movement, where $\left(x\right)^{[k]}$ denotes the k^{th} iterate of an estimate of a quantity *x*.

i. In general, population recruitment by area and gender is a function of area-specific recruitment production and the recruitment distribution matrix $\underline{\underline{Q}}^{(p)} = \left(Q_{i \to j}^{(p)}\right)_{d\,x\,d}$. Set age-0 recruits as a function of unfished recruitment and recruitment distribution by area and gender via

$$\left(N_{unfished,d,a=0,g}^{(p)}\right)^{[1]} = \rho_g^{(p)} \sum_k R_{unfished,k}^{(p)} \cdot Q_{k \to d}^{(p)}$$

ii. Set age-*a* survivors by area and gender for ages *a*=1 to *A*-1 via

$$\left(N_{unfished,d,a,g}^{(p)}\right)^{[1]} = \left(N_{unfished,d,a-1,g}^{(p)}\right)^{[1]} \cdot \exp\left(-M_{unfished,a-1,g}^{(p)}\right)$$

iii. Set age-*A* group of survivors by area and gender via

$$\left(N_{unfished,d,A,g}^{(p)}\right)^{[1]} = \frac{\left(N_{unfished,d,A-1,g}^{(p)}\right)^{[1]} \cdot \exp\left(-M_{unfished,A-1,g}^{(p)}\right)}{1-\exp\left(-M_{unfished,A,g}^{(p)}\right)}$$

iv. Set unfished spawning biomass by population, area and gender via

$$\left(SB^{(p)}_{unfished,d,g}\right)^{[1]} = \sum_{a} P^{(p)}_{mature,a,g} \cdot W^{(p)}_{spawn,a,g} \cdot \left(N^{(p)}_{unfished,d,a,g}\right)^{[1]} \cdot \exp\left(-\Delta_S \cdot M^{(p)}_{unfished,a,g}\right)$$

Iteration i=2: Calculate the next iterate of unfished numbers at age estimates by population, area, and gender based on unfished recruitment, the recruitment distribution, equilibrium survival, movement probabilities by area, age, and gender, and the previous iterate. Unfished recruitment production by area is a function of area-specific spawning biomasses which need to be iteratively calculated to account for the movement probabilities that redistribute fish.

i. Set age-0 recruits as a function of unfished recruitment and recruitment distribution by area and gender (Note that this calculation of unfished recruitment by population, area and gender does not change between iterations and can be done once, but is listed here to show the iterative process and emphasize the dependence on the unfished recruitment parameters by population and area, $R^{(p)}_{unfished,k}$) via

$$\left(N^{(p)}_{unfished,d,a=0,g}\right)^{[2]} = \left(N^{(p)}_{unfished,d,a=0,g}\right)^{[1]} \equiv \rho^{(p)}_{g} \sum_{k} R^{(p)}_{unfished,k} \cdot Q^{(p)}_{k \to d}$$

ii. Set age-*a* survivors for ages *a*=1 to *A*-1 by population, area, and gender that did not emigrate plus age-*a* surviving immigrants from other areas via

$$\left(N^{(p)}_{unfished,d,a,g}\right)^{[2]} = \sum_{k} \left(N^{(p)}_{unfished,k,a-1,g}\right)^{[1]} \cdot \exp\left(-M^{(p)}_{unfished,a-1,g}\right) \cdot T^{(p)}_{k \to d,a-1,g}$$

iii. Set age-(*A*-1) survivors that did not emigrate plus age-(*A*-1) immigrants from other areas plus age-*A* group survivors that did not emigrate plus age-*A* group immigrants from other areas via

$$\begin{aligned}\left(N^{(p)}_{unfished,d,A,g}\right)^{[2]} &= \sum_{k} \left(N^{(p)}_{unfished,k,A-1,g}\right)^{[1]} \cdot \exp\left(-M^{(p)}_{unfished,A-1,g}\right) \cdot T^{(p)}_{k \to d,A-1,g} \\ &\quad + \sum_{k} \left(N^{(p)}_{unfished,k,A,g}\right)^{[1]} \cdot \exp\left(-M^{(p)}_{unfished,A,g}\right) \cdot T^{(p)}_{k \to d,A,g}\end{aligned}$$

v. Set unfished spawning biomass by population, area and gender via

$$\left(SB^{(p)}_{unfished,d,g}\right)^{[2]} = \sum_{a} P^{(p)}_{mature,a,g} \cdot W^{(p)}_{spawn,a,g} \cdot \left(N^{(p)}_{unfished,d,a,g}\right)^{[2]} \cdot \exp\left(-\Delta_R \cdot M^{(p)}_{unfished,a,g}\right)$$

Iteration i=j+1: Calculate the (*j*+1)st iterate of equilibrium fished numbers at age estimates by population, area, and gender based on unfished recruitment, the recruitment distribution, equilibrium survival, movement probabilities by area, age, and gender, and the *j*th iterate.

i. Set age-0 recruits by population, area, and gender via

$$\left(N_{unfished,d,a=0,g}^{(p)}\right)^{[j+1]} = \left(N_{unfished,d,a=0,g}^{(p)}\right)^{[j]}$$

ii. Set age-*a* survivors for ages *a*=1 to *A*-1 by population, area, and gender that did not emigrate plus age-*a* surviving immigrants from other areas via

$$\left(N_{unfished,d,a,g}^{(p)}\right)^{[j+1]} = \sum_k \left(N_{unfished,k,a-1,g}^{(p)}\right)^{[j]} \cdot \exp\left(-M_{unfished,a-1,g}^{(p)}\right) \cdot T_{k\to d,a-1,g}^{(p)}$$

iii. Set age-(*A*-1) survivors that did not emigrate plus age-(*A*-1) immigrants from other areas plus age-*A* group survivors that did not emigrate plus age-*A* group immigrants from other areas via

$$\begin{aligned}\left(N_{unfished,d,A,g}^{(p)}\right)^{[j+1]} &= \sum_k \left(N_{unfished,k,A-1,g}^{(p)}\right)^{[j]} \cdot \exp\left(-M_{unfished,A-1,g}^{(p)}\right) \cdot T_{k\to d,A-1,g}^{(p)} \\ &+ \sum_k \left(N_{unfished,k,A,g}^{(p)}\right)^{[j]} \cdot \exp\left(-M_{unfished,A,g}^{(p)}\right) \cdot T_{k\to d,A,g}^{(p)}\end{aligned}$$

vi. Set unfished spawning biomass by population, area and gender via

$$\left(SB_{unfished,d,g}^{(p)}\right)^{[j+1]} = \sum_a P_{mature,a,g}^{(p)} \cdot W_{spawn,a,g}^{(p)} \cdot \left(N_{unfished,d,a,g}^{(p)}\right)^{[j+1]} \cdot \exp\left(-\Delta_S \cdot M_{unfished,a,g}^{(p)}\right)$$

Continue the iterations until the convergence criteria below is achieved or the maximum number of iterations has been reached.

Convergence Criterion

Calculate the distance between successive sets of unfished equilibrium spawning biomass estimates by population, area and gender, denoted by $D^{[j]}$, by applying the L_1, or least absolute deviations norm, to the set of estimates as

$$D^{[j]} = \sum_p \sum_d \sum_g \left| \left(SB_{unfished,d,g}^{(p)}\right)^{[j+1]} - \left(SB_{unfished,d,g}^{(p)}\right)^{[j]} \right|$$

Stop the iterations when the set of unfished spawning biomass estimates have converged. That is, stop when $D^{[j]} < \varepsilon \; \; for \; a \; \; small \; value \; \varepsilon > 0$.

If the iterations converge, then one has determined the unfished numbers at age by population, area, and gender $\left(N^{(p)}_{unfished,d,a,g}\right)$ along with the unfished spawning biomass by population, area, and gender $\left(SB^{(p)}_{unfished,d,g}\right)$.

2.2 Equilibrium Fished Numbers at Age Algorithm

Similarly, one needs to calculate fished numbers at age in equilibrium prior to the start of the assessment time horizon by population, area, and gender $\left(N^{(p)}_{fished,d,a,g}\right)$ as a function of population recruitment distribution by area, movement probabilities, and the equilibrium total mortality at age. The equilibrium fished numbers at age by population, area, and gender depend on the population movement and recruitment distribution matrices and are needed to compute the fished equilibrium spawning biomasses by population and area, which in turn, are needed to calculate the equilibrium numbers at by population, area, and gender for the initial fished equilibrium time period to the start of the assessment time horizon.

<u>Iteration i=1</u>: Calculate the initial equilibrium fished numbers at age estimates by population, area, and gender <u>based on unfished recruitment, the recruitment distribution, equilibrium total mortality and</u> <u>no movement,</u> where $\left(x\right)^{[j]}$ denotes the *j*th iterate of an estimate of a quantity *x*.

i. In general, population recruitment by area and gender is a function of area-specific recruitment production and the recruitment distribution matrix $\underline{\underline{Q}}^{(p)} = \left(Q^{(p)}_{i \to j}\right)_{d\,x\,d}$. Set the initial age-0 fished recruits as a function of unfished recruitment and recruitment distribution by area and gender via

$$\left(N^{(p)}_{fished,d,a=0,g}\right)^{[1]} \equiv \left(R^{(p)}_{fished,k,g}\right)^{[1]} = \rho^{(p)}_{g} \sum_{k} Q^{(p)}_{k \to d} \cdot R^{(p)}_{unfished,k}$$

ii. Set initial age-*a* survivors by area and gender for ages *a*=1 to *A*-1 from the initial fished recruits and equilibrium total mortality by area and gender via

$$\left(N^{(p)}_{fished,d,a,g}\right)^{[1]} = \left(N^{(p)}_{fished,d,a-1,g}\right)^{[1]} \cdot \exp\left(-Z^{(p)}_{fished,d,a-1,g}\right)$$

iii. Set initial Age-*A* group of survivors from the initial fished recruits and equilibrium total mortality by area and gender via

$$\left(N^{(p)}_{fished,d,A,g}\right)^{[1]} = \frac{\left(N^{(p)}_{fished,d,A-1,g}\right)^{[1]} \cdot \exp\left(-Z^{(p)}_{fished,d,A-1,g}\right)}{1-\exp\left(-Z^{(p)}_{fished,d,A,g}\right)}$$

iv. Set equilibrium fished spawning biomass by population, area and gender via

$$\left(SB^{(p)}_{fished,d,g}\right)^{[1]} = \sum_a P^{(p)}_{mature,a,g} \cdot W^{(p)}_{spawn,a,g} \cdot \left(N^{(p)}_{fished,d,a,g}\right)^{[1]} \cdot \exp\left(-\Delta_S \cdot Z^{(p)}_{fished,d,a,g}\right)$$

Iteration i=2: Calculate the next iterate of the equilibrium fished numbers at age estimates by population, area, and gender based on the recruitment submodel, the recruitment distribution, equilibrium total mortality, movement probabilities by area, age, and gender, and the previous iterate. Equilibrium fished recruitment production by area is a function of area-specific spawning biomasses which need to be iteratively calculated to account for the movement probabilities that redistribute fish.

i. Set age-0 recruits as a function of the recruitment submodel $f^{(p)}_d$ and recruitment distribution by area and gender (Note that this is effectively turning on the recruitment dynamics) via

$$\left(R^{(p)}_{fished,d}\right)^{[2]} = f^{(p)}_d\left(\left(SB^{(p)}_{fished,d,g=female}\right)^{[1]} \mid \underline{\theta}^{(p)}_d \text{ , } SB^{(p)}_{unfished,d,g=female}\right) \text{ and}$$

$$\left(N^{(p)}_{fished,d,a=0,g}\right)^{[2]} = \rho^{(p)}_g \sum_k Q^{(p)}_{k\to d} \cdot \left(R^{(p)}_{fished,k}\right)^{[2]}$$

ii. Set age-*a* survivors for ages *a*=1 to *A*-1 by population, area, and gender as survivors that did not emigrate plus age-*a* surviving immigrants from other areas (Note that this is effectively turning on the movement dynamics) via

$$\left(N^{(p)}_{fished,d,a,g}\right)^{[2]} = \sum_k \left(N^{(p)}_{fished,k,a-1,g}\right)^{[1]} \cdot \exp\left(-Z^{(p)}_{fished,k,a-1,g}\right) \cdot T^{(p)}_{k\to d,a-1,g}$$

iii. Set age-(*A*-1) survivors that did not emigrate plus age-(*A*-1) immigrants from other areas plus age-*A* group survivors that did not emigrate plus age-*A* group immigrants from other areas via

$$\begin{aligned}\left(N^{(p)}_{fished,d,A,g}\right)^{[2]} = &\sum_k \left(N^{(p)}_{fished,k,A-1,g}\right)^{[1]} \cdot \exp\left(-Z^{(p)}_{fished,k,A-1,g}\right) \cdot T^{(p)}_{k\to d,A-1,g} \\ &+ \sum_k \left(N^{(p)}_{fished,k,A,g}\right)^{[1]} \cdot \exp\left(-Z^{(p)}_{fished,k,A,g}\right) \cdot T^{(p)}_{k\to d,A,g}\end{aligned}$$

iv. Set equilibrium fished spawning biomass by population, area and gender via

$$\left(SB^{(p)}_{fished,d,g}\right)^{[2]} = \sum_a P^{(p)}_{mature,a,g} \cdot W^{(p)}_{spawn,a,g} \cdot \left(N^{(p)}_{fished,d,a,g}\right)^{[2]} \cdot \exp\left(-\Delta_S \cdot Z^{(p)}_{fished,d,a,g}\right)$$

<u>Iteration i=j+1</u>: Calculate the (*kj*1)st iterate of equilibrium fished numbers at age estimates by population, area, and gender based on the recruitment submodels and recruitment distribution, equilibrium total mortality, movement probabilities by area, age, and gender, and the *j*th iterate.

i. Set age-0 recruits as a function of the recruitment submodel $f^{(p)}_d$ and recruitment distribution by area and gender via

$$\left(R^{(p)}_{fished,d}\right)^{[j+1]} = f^{(p)}_d\left(\left(SB^{(p)}_{fished,d,g=female}\right)^{[j]} \mid \underline{\theta}^{(p)}_d,\ SB^{(p)}_{unfished,d,g=female}\right) \text{ and}$$

$$\left(N^{(p)}_{fished,d,a=0,g}\right)^{[j+1]} = \rho^{(p)}_g \sum_k Q^{(p)}_{k\to d} \cdot \left(R^{(p)}_{fished,k}\right)^{[j]}$$

ii. Set age-*a* survivors for ages *a*=1 to *A*-1 by population, area, and gender that did not emigrate plus age-*a* surviving immigrants from other areas via

$$\left(N^{(p)}_{fished,d,a,g}\right)^{[j+1]} = \sum_k \left(N^{(p)}_{fished,k,a-1,g}\right)^{[j]} \cdot \exp\left(-Z^{(p)}_{fished,k,a-1,g}\right) \cdot T^{(p)}_{k\to d,a-1,g}$$

iii. Set age-(*A*-1) survivors that did not emigrate plus age-(*A*-1) immigrants from other areas plus age-*A* group survivors that did not emigrate plus age-*A* group immigrants from other areas via

$$\begin{aligned}\left(N^{(p)}_{fished,d,A,g}\right)^{[j+1]} = &\sum_k \left(N^{(p)}_{fished,k,A-1,g}\right)^{[j]} \cdot \exp\left(-Z^{(p)}_{unfished,k,A-1,g}\right) \cdot T^{(p)}_{k\to d,A-1,g} \\ &+ \sum_k \left(N^{(p)}_{fished,k,A,g}\right)^{[j]} \cdot \exp\left(-Z^{(p)}_{unfished,k,A,g}\right) \cdot T^{(p)}_{k\to d,A,g}\end{aligned}$$

v. Set unfished spawning biomass by population, area and gender via

$$\left(SB^{(p)}_{fished,d,g}\right)^{[j+1]} = \sum_a P^{(p)}_{mature,a,g} \cdot W^{(p)}_{spawn,a,g} \cdot \left(N^{(p)}_{fished,d,a,g}\right)^{[j+1]} \cdot \exp\left(-\Delta_S \cdot Z^{(p)}_{fished,d,a,g}\right)$$

Continue the iterations until convergence is achieved or the maximum number of iterations has been reached.

If the algorithm converges, then one has determined the unfished numbers at age by population, area, and gender $\left(N^{(p)}_{fished,d,a,g}\right)$ along with the unfished spawning biomass by population, area, and gender $\left(SB^{(p)}_{fished,d,g}\right)$. This population-specific information sets the initial conditions at the start (first year) of the initialization time period, prior to the stock assessment time horizon. These initial conditions, along with recruitment deviation parameters, determine the population dynamics for the initialization time period.

2.3 Dynamic Fished Numbers at Age Algorithm

2.3.1 Set Initial Numbers at Age

Set initial numbers at age in year *y*=1 of the assessment time horizon equal to the equilibrium fished numbers at age by population, area and gender as $N^{(p)}_{y=1,d,a,g} = N^{(p)}_{fished,d,a,g}$

Loop over years (*y*=1,…,*Y*)

2.3.2 Calculate predicted fishery selectivities at age by population, fleet, and gender

$$S^{(p)}_{v,g}(a) = \frac{1}{1+\exp\left(-\frac{\left(a-a^{(p)}_{50,v,g}\right)}{\sigma^{(p)}_{v,g}}\right)}$$

2.3.3 Calculate predicted survey selectivities at age by population, fleet, and gender

$$S^{(p)}_{s,g}(a) = \frac{1}{1+\exp\left(-\frac{\left(a-a^{(p)}_{50,s,g}\right)}{\sigma^{(p)}_{s,g}}\right)}$$

2.3.4 Calculate predicted total mortality rates at age by year, population, fleet, and gender

$$Z^{(p)}_{y,d,a,g} = M^{(p)}_{d,a,g} + S^{(p)}_{v,g}\left(a\right) F^{(p)}_{v,y,d}$$

2.3.5 Calculate predicted female spawning biomass by year, population, and area

$$SB^{(p)}_{y,d,g=female} = \sum_a P^{(p)}_{mature,a,g=female} \cdot W^{(p)}_{spawn,a,g=female} \cdot N^{(p)}_{y,d,a,g=female} \cdot \exp\left(-\Delta_S Z^{(p)}_{y,d,a,g=female}\right)$$

Where $N^{(p)}_{y,d,a,g=female} \cdot \exp\left(-\Delta_S Z^{(p)}_{y,d,a,g=female}\right)$ is the number of surviving adult female spawners

2.3.6 Calculate predicted recruitment by year, population, and area

$$R^{(p)}_{y,d,g} = \rho^{(p)}_g \left(Q^{(p)}_{d\to d} R^{(p)}_{y,d} + \sum_{k\neq d} Q^{(p)}_{k\to d} R^{(p)}_{y,k} \right)$$

Where $R^{(p)}_{y,d} = \dfrac{4h^{(p)}_d \cdot R^{(p)}_{unfished,d} \cdot SB^{(p)}_{y,d}}{SB^{(p)}_{unfished,d}\left(1-h^{(p)}_d\right) + SB^{(p)}_{y,d}\left(5h^{(p)}_d - 1\right)}$ is the number of recruitment produced by year, population and area, and

Where $\underline{\underline{Q}}^{(p)} = \left(Q^{(p)}_{i\to j}\right)$ is the recruitment distribution by area matrix for each population, and

Where $N^{(p)}_{y,d,a=0,g} = R^{(p)}_{y,d,g}$ is the recruitment strength by population, area and gender in year *y*.

2.3.7 Calculate predicted fishery observations

Calculate predicted fishery catch numbers at age by fleet, year, population, area, and gender as

$$C^{(p)}_{v,y,d,a,g} = N^{(p)}_{y,d,a,g} \frac{F^{(p)}_{v,y,d} S^{(p)}_{v,d,g}\left(a\right)}{\left(F^{(p)}_{v,y,d} S^{(p)}_{v,d,g}\left(a\right) + M^{(p)}_{d,a,g}\right)} \left\{1 - \exp\left[-\left(F^{(p)}_{v,y,d} S^{(p)}_{v,d,g}\left(a\right) + M^{(p)}_{d,a,g}\right)\right]\right\}$$

Calculate predicted fishery catch proportion at age by fleet, year, population, area, and gender as

$$P^{(p)}_{v,y,d,a,g} = \frac{C^{(p)}_{v,y,d,a,g}}{\sum_a C^{(p)}_{v,y,d,a,g}}$$

Calculate predicted fishery catch biomass at age by fleet, year, population, area, and gender as

$$CB^{(p)}_{v,y,d,a,g} = C^{(p)}_{v,y,d,a,g} \cdot W^{(p)}_{v,d,a,g}$$

2.3.8 Calculate predicted survey observations

Calculate predicted survey catch numbers at age by year, population, area, and gender as

$$C^{(p)}_{I,y,d,a,g} = S^{(p)}_{I,d,a,g} N^{(p)}_{y,d,a,g} \cdot \exp\left(-\Delta_I \cdot Z^{(p)}_{y,d,a,g}\right)$$

Calculate predicted survey catch proportion at age by year, population, area, and gender as

$$P^{(p)}_{I,y,d,a,g} = \frac{C^{(p)}_{I,y,d,a,g}}{\sum_a C^{(p)}_{I,y,d,a,g}}$$

Calculate predicted survey catch biomass at age by fleet, year, population, area, and gender as

$$CB^{(p)}_{I,y,d,a,g} = q_{I,d} \cdot C^{(p)}_{I,y,d,a,g} \cdot W^{(p)}_{I,d,a,g}$$

Where $q_{I,d}$ is the catchability of survey *I* in area *d*

2.3.9 Calculate predicted quantities of interest

Calculate predicted total fishery catch numbers at age summed over populations by fleet, year, area, and gender as

$$C_{v,y,d,a,g} = \sum_p C^{(p)}_{v,y,d,a,g}$$

Calculate predicted total fishery catch proportion at age summed over populations by fleet, year, area, and gender as

$$P_{v,y,d,a,g} = \frac{\sum_p C^{(p)}_{v,y,d,a,g}}{\sum_p \sum_a C^{(p)}_{v,y,d,a,g}}$$

Calculate predicted total fishery catch biomass summed over populations by fleet, year, and area as

$$CB_{v,y,d} = \sum_p \sum_a \sum_g CB^{(p)}_{v,y,d,a,g}$$

Calculate predicted total survey catch numbers at age summed over populations by year, area, and gender as

$$C_{I,y,d,a,g} = \sum_{p} C_{I,y,d,a,g}^{(p)}$$

Calculate predicted survey catch proportion at age by year, area, and gender as

$$P_{I,y,d,a,g} = \frac{\sum_{p} C_{I,y,d,a,g}^{(p)}}{\sum_{p}\sum_{a} C_{I,y,d,a,g}^{(p)}}$$

Calculate predicted total survey catch biomass by fleet, year, and area as

$$CB_{I,y,d} = \sum_{p}\sum_{a}\sum_{g} CB_{I,y,d,a,g}^{(p)}$$

<u>2.3.10 Calculate population numbers at age at the start of year *y*+1</u>

Set age-*a* survivors for ages *a*=1 to *A*-1 by population, area, and gender that did not emigrate plus age-*a* surviving immigrants from other areas via

$$N_{y+1,d,a,g}^{(p)} = \sum_{k} N_{y,k,a-1,g}^{(p)} \cdot \exp\left(-Z_{y,k,a-1,g}^{(p)}\right) \cdot T_{k\to d,a-1,g}^{(p)}$$

Set age-(*A*-1) survivors that did not emigrate plus age-(*A*-1) immigrants from other areas plus age-*A* group survivors that did not emigrate plus age-*A* group immigrants from other areas via

$$\begin{aligned} N_{y+1,d,A,g}^{(p)} &= \sum_{k} N_{y,k,A-1,g}^{(p)} \cdot \exp\left(-Z_{y,k,A-1,g}^{(p)}\right) \cdot T_{k\to d,A-1,g}^{(p)} \\ &\quad + \sum_{k} N_{y,k,A,g}^{(p)} \cdot \exp\left(-Z_{y,k,A,g}^{(p)}\right) \cdot T_{k\to d,A,g}^{(p)} \end{aligned}$$

3 Write Results to Output File

The input data and output results of applying the equilibrium and time series algorithms for case study 1 (Table 11) are written to the output file named **case_study_v1_output.txt** (Table 12).

Table 11. Case study 1 algorithm.

```
1. Initialize Model
    1.1. Initialize Analysis Class
    1.2. Set Model Domain Parameters
    1.3. Initialize Population Class
    1.4. Initialize Observation Class
    1.5. Initialize Environment Class
    1.6. Set Initial Parameter Vector
    1.7. Write Input Data to Output File
2. Calculate Population Size Through Time
    2.1. Calculate Unfished Equilibrium Numbers at Age
        2.1.1.  Calculate Initial Unfished Numbers at Age Iterate
            WHILE (HasConverged = FALSE) AND (Iteration < Maximum Iteration)
        2.1.2.  Calculate Next Unfished Numbers at Age Iterate
        2.1.3.  Calculate Convergence Criterion
        2.1.4.  IF Convergence Criterion Achieved THEN Set HasConverged = TRUE
            END WHILE
        2.1.5.  Write Unfished Equilibrium Results to Output File
        2.1.6.  Update Initial Parameter Vector
    2.2. Calculate Fished Equilibrium Numbers at Age
        2.2.1.  Calculate Initial Fished Numbers at Age Iterate
            WHILE (HasConverged = FALSE) AND (Iteration< Maximum Iteration)
        2.2.2.  Calculate Next Unfished Numbers at Age Iterate
        2.2.3.  Calculate Convergence Criterion
        2.2.4.  IF Convergence Criterion Achieved THEN Set HasConverged = TRUE
            END WHILE
        2.2.5.  Write Fished Equilibrium Results to Output File
    2.3. Calculate Dynamic Numbers at Age in Assessment Time Horizon
        2.3.1.  Set Current Parameter Vector Equal to Initial Parameter Vector
            FOR (year $y$=1 to $Y$)
        2.3.2.  Calculate Predicted Fishery Selectivities
        2.3.3.  Calculate Predicted Survey Selectivities
        2.3.4.  Calculate Predicted Total Mortality Rates
        2.3.5.  Calculate Predicted Spawning Biomass
        2.3.6.  Calculate Predicted Recruitment By Population
        2.3.7.  Calculate Predicted Fishery Observations
        2.3.8.  Calculate Predicted Survey Observations
        2.3.9.  Calculate Predicted Quantities of Interest By Population
        2.3.10. Calculate Population Numbers at Age at the Start of Year $y$+1
        2.3.11. Calculate Population Mean Lengths and Weights at Age for the Plus Group
```

3. Write Model Results to Output File

3.1.1. Predicted Fishery Selectivities

3.1.2. Predicted Survey Selectivities

3.1.3. Predicted Fishing Mortality Rates

3.1.4. Predicted Total Mortality Rates

3.1.5. Calculate Predicted Total Mortality Rates

3.1.6. Predicted Spawning Biomass

3.1.7. Predicted Recruitment By Population

3.1.8. Predicted Fishery Observations

3.1.9. Predicted Survey Observations

3.1.10. Predicted Quantities of Interest By Population

3.1.11. Predicted Population Numbers at Age at the Start of the Year

3.1.12. Predicted Population Mean Lengths and Weights at Age for the Plus Group at the Start of the Year

Table 12. Glossary of quantities of interest, R language variable names and population equations. Array index labels are: “*p*” for population, “*d*” for area, “*g*” for gender, “*a*” for age, “*v*” for fleet, “*I*” is for survey and “*y*” for year.

Quantity of Interest	Variable Name in R	Population Dynamics Equation
Predicted fishery selectivity at age by population, area/fleet, and gender	FisherySelectivityAtAge[p,d,g,a]	$S_{v,g}^{(p)}(a)=\left\{1+\exp\left(-\frac{\left(a-a_{50,v,g}^{(p)}\right)}{\sigma_{v,g}^{(p)}}\right)\right\}^{-1}$
Predicted survey selectivity at age by population, area/fleet, and gender	SurveySelectivityAtAge[p,d,g,a]	$S_{I,d,g}^{(p)}(a)=\left\{1+\exp\left(-\frac{\left(a-a_{50,I,d,g}^{(p)}\right)}{\sigma_{I,d,g}^{(p)}}\right)\right\}^{-1}$
Predicted fishing mortality rate (fully-selected) by year, fleet and area	FishingMortality[y,v,d]	$F_{v,y,d}$ are fixed inputs defined in Table 1
Predicted total mortality rate by year, population, area, gender and age	TotalMortality[y,p,d,g,a]	$Z_{y,d,a,g}^{(p)}$ are constants defined by fixed natural mortality, fishing mortality and fishery selectivity values
Predicted spawning biomass by year, population, area and gender	SpawningBiomass[y,p,d,g]	$SB_{y,d,g=female}^{(p)}=\sum_{a}P_{mature,a,g=female}^{(p)}\cdot W_{spawn,a,g=female}^{(p)}\cdot N_{y,d,a,g=female}^{(p)}\cdot\exp\left(-\Delta_{S}Z_{y,d,a,g=female}^{(p)}\right)$

Table 11. Continued.

Quantity of Interest	Variable Name in R	Population Dynamics Equation
Predicted recruitment (female and male recruits) by year, population and area	Recruitment[y,p,d,1]+Recruitment[y,p,d,2]	$R^{(p)}_{y,d} = Q^{(p)}_{d\to d} R^{(p)}_{y,d} + \sum_{k\neq d} Q^{(p)}_{k\to d} R^{(p)}_{y,k}$
Predicted fishery catch numbers at age by fleet, year, population, area and gender	FisheryCatchNumbersAtAge[y,p,v,d,g,a]	$C^{(p)}_{v,y,d,a,g} = N^{(p)}_{y,d,a,g} \frac{F^{(p)}_{v,y,d} S^{(p)}_{v,d,g}(a)}{\left(F^{(p)}_{v,y,d} S^{(p)}_{v,d,g}(a) + M^{(p)}_{d,a,g}\right)} \left\{1-\exp\left[-\left(F^{(p)}_{v,y,d} S^{(p)}_{v,d,g}(a) + M^{(p)}_{d,a,g}\right)\right]\right\}$
Predicted fishery catch proportions at age by fleet, year, area and gender	FisheryCatchProportionAtAge[y,p,v,d,g,a]	$P_{v,y,d,a,g} = \frac{\sum_{p} C^{(p)}_{v,y,d,a,g}}{\sum_{p}\sum_{a} C^{(p)}_{v,y,d,a,g}}$
Predicted fishery catch biomass at age by fleet, year, area and gender	FisheryCatchBiomassAtAge[y,p,v,d,g,a]	$CB^{(p)}_{v,y,d,a,g} = C^{(p)}_{v,y,d,a,g} \cdot W^{(p)}_{v,d,a,g}$
Predicted survey catch numbers at age by year, population, area and gender	SurveyCatchNumbersAtAge[y,p,d,g,a]	$C^{(p)}_{I,y,d,a,g} = S^{(p)}_{I,d,a,g} N^{(p)}_{y,d,a,g} \cdot \exp\left(-\Delta_I \cdot Z^{(p)}_{y,d,a,g}\right)$

Table 11. Continued.

Quantity of Interest	Variable Name in R	Population Dynamics Equation
Predicted survey catch proportions at age by year, area and gender	SurveyCatchProportionAtAge[y,p,d,g,a]	$P_{I,y,d,a,g}^{(p)} = \frac{C_{I,y,d,a,g}^{(p)}}{\sum_a C_{I,y,d,a,g}^{(p)}}$
Predicted survey catch biomass at age by year, area and gender	SurveyCatchBiomassAtAge[y,p,d,g,a]	$CB_{I,y,d,a,g}^{(p)} = q_{I,d} \cdot C_{I,y,d,a,g}^{(p)} \cdot W_{I,d,a,g}^{(p)}$
Predicted total fishery catch numbers at age summed over populations by fleet, year, area, and gender	PredictedFisheryCatchNumbersAtAge[y,v,d,g,a]	$C_{v,y,d,a,g} = \sum_p C_{v,y,d,a,g}^{(p)}$
Predicted total fishery catch proportion at age summed over populations by fleet, year, area, and gender	PredictedFisheryCatchProportionAtAge[y,v,d,g,a]	$P_{v,y,d,a,g} = \frac{\sum_p C_{v,y,d,a,g}^{(p)}}{\sum_p \sum_a C_{v,y,d,a,g}^{(p)}}$
Predicted total fishery catch biomass summed over populations by fleet, year, and area	PredictedFisheryCatchBiomass[y,v,d]	$CB_{v,y,d} = \sum_p \sum_a \sum_g CB_{v,y,d,a,g}^{(p)}$
Predicted total survey catch numbers at age summed over populations by year, area, and gender	PredictedSurveyCatchNumbersAtAge[y,d,g,a]	$C_{I,y,d,a,g} = \sum_p C_{I,y,d,a,g}^{(p)}$

Table 11. Continued.

Quantity of Interest	Variable Name in R	Population Dynamics Equation
Predicted total survey catch proportion at age summed over populations year, area, and gender	PredictedSurveyCatchProportionAtAge[y,d,g,a]	$P_{I,y,d,a,g} = \frac{\sum_{p} C^{(p)}_{I,y,d,a,g}}{\sum_{p}\sum_{a} C^{(p)}_{I,y,d,a,g}}$
Predicted total survey catch biomass summed over populations by year, and area	PredictedSurveyCatchBiomass[y,d]	$CB_{I,y,d} = \sum_{p}\sum_{a}\sum_{g} CB^{(p)}_{I,y,d,a,g}$
Predicted population numbers at age on January 1st by year, area and gender	NumbersAtAge[y,p,d,g,a]	For true ages $N^{(p)}_{y+1,d,a,g} = \sum_{k} N^{(p)}_{y,k,a-1,g} \cdot \exp\left(-Z^{(p)}_{y,k,a-1,g}\right) \cdot T^{(p)}_{k\rightarrow d,a-1,g}$ For the plus group $N^{(p)}_{y+1,d,A,g} = \sum_{k} N^{(p)}_{y,k,A-1,g} \cdot \exp\left(-Z^{(p)}_{y,k,A-1,g}\right) \cdot T^{(p)}_{k\rightarrow d,A-1,g}$ $+ \sum_{k} N^{(p)}_{y,k,A,g} \cdot \exp\left(-Z^{(p)}_{y,k,A,g}\right) \cdot T^{(p)}_{k\rightarrow d,A,g}$

Appendix 1. R script (case_study_v1_20180316.R) to run simulation calculations from the current working directory with file input and output for MAS Case Study 1.

```
###############################################################################################################
# The R program file case_study_v1_20180316.R implements data input
# and initial calculations for case study v1 of the MAS project.
# Jon Brodziak, PIFSC, jon.brodziak@noaa.gov
###############################################################################################################
# 3-Oct-2017
# Removed 10-year fixed initialization time period and associated deprecated code.
# 6-Oct-2017
# Added function to write input data to output file.
# 18-Dec-2017
# Completed time loop to calculate numbers at age array and quantities of interest.
# 14-Feb-2018
# Added assessment time period results to output file.
# 16-Mar-2018
# Completed scripts to read and echo input data and revised initialization processes
# and implemented the equilibrium and time-varying MaxPlusGroupAge adjustments.
#
###############################################################################################################
#
# (1) MODEL INITIALIZATION
#
###############################################################################################################
# (1.1) INITIALIZE ANALYSIS CLASS
###############################################################################################################
# NOTE the age groups in each population consist of age=0 to age=9+
# and the age group information is stored in array positions a=1 to a=10.
# NOTE that females are indexed by g=1 and males by g=2 in arrays.
# NOTE that fleet 1 fishes in area 1 and fleet 2 fishes in area 2.
# NOTE that survey 1 operates in area 1 and survey 2 operates in area 2.
#----------------------------------------------------------------------------------------------

# Remove all variable objects from the session memory
rm(list=ls(all.names=F))

# Close any open graphics windows
graphics.off()

# workstation setup for source.folder
source.folder <- 'D:/Data/MAS/case.study/code/'

# laptop setup for source.folder
# source.folder <- 'C:/Users/Jon.Brodziak/Desktop/MAS/case.study/code/'

source(paste(source.folder,'Maturity_Logistic.R',sep=""))

source(paste(source.folder,'Growth_MVB.R',sep=""))

source(paste(source.folder,'WeightAtLength_Allometric.R',sep=""))

source(paste(source.folder,'Recruitment_BH.R',sep=""))

source(paste(source.folder,'Selectivity_Logistic.R',sep=""))

source(paste(source.folder,'Write_Input_Data.R',sep=""))

################################################################################################
# (1.2) SET MODEL DOMAIN PARAMETERS
################################################################################################

source(paste(source.folder,'Read_Input_Data.R',sep=""))
```

```
###############################################################################################
# (1.3) INITIALIZE POPULATION CLASS
###############################################################################################

# Compute maturity probabilities at age by gender, area, and population
#--------------------------------------------------------------------------------------------
MaturityProbabilityAtAge <- array(rep(1.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))

for (p in 1:NPopulation)
  for (d in 1:NArea)
    for (g in 1:NGender)
      {
      for (a in 1:NAge)
        {
        age <- a-1
        MaturityProbabilityAtAge[p,d,g,a]  <- Maturity.Logistic(age,Maturity.a50[p,d,g],Maturity.slope[p,d,g])
        }
      }

# Compute mean length at age at start of year by gender, area, and population
#--------------------------------------------------------------------------------------------
MeanLengthStartOfYear <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))

for (p in 1:NPopulation)
  for (d in 1:NArea)
    for (g in 1:NGender)
      for (a in 2:(NAge-1)) {
        age <- a-1
        MeanLengthStartOfYear[p,d,g,a]  <- Growth.MVB(age,Length.Amin[p,d,g],Length.Amax[p,d,g],StartZFraction[p,d],Length.Lmin[p,d,g],Length.Lmax[p,d,g],Length.c[p,d,g])
      }
```

```
# Use linear interpolation for age-0 fish (index a=1)

for (p in 1:NPopulation)

  for (d in 1:NArea)

    for (g in 1:NGender)

    {

      a <- 1

      MeanLengthStartOfYear[p,d,g,a]  <-  StartZFraction[p,d]*MeanLengthStartOfYear[p,d,g,(a+1)]

    }

# Use MaxPlusGroup age adjustment for the plus group

for (p in 1:NPopulation)

  for (d in 1:NArea)

    for (g in 1:NGender)

    {

      tmp <- 0.0

      for (k in 0:(MaxPlusGroupAge-NAge-1))

        tmp <- tmp + exp(-k*NaturalMortality[p,d,g,NAge])*Growth.MVB((NAge+k),Length.Amin[p,d,g],Length.Amax[p,d,g],StartZFraction[p,d],Length.Lmin[p,d,g],Length.Lmax[p,d,g],Length.c[p,d,g])

      tmp <- tmp*(1.0-exp(-NaturalMortality[p,d,g,NAge]))

      MeanLengthStartOfYear[p,d,g,NAge] <- tmp/(1.0-exp(-NaturalMortality[p,d,g,NAge]*(MaxPlusGroupAge-NAge)))

    }


# Compute mean length during spawning season age by gender, area, and population

#-------------------------------------------------------------------------------------------------

MeanLengthSpawning <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))



for (p in 1:NPopulation)

  for (d in 1:NArea)

    for (g in 1:NGender)

      for (a in 2:(NAge-1)) {

        age <- a-1
        MeanLengthSpawning[p,d,g,a]  <- Growth.MVB(age,Length.Amin[p,d,g],Length.Amax[p,d,g],SpawningZFraction[p,d],Length.Lmin[p,d,g],Length.Lmax[p,d,g],Length.c[p,d,g])

      }

# Use linear interpolation for age-0 fish (index a=1)

for (p in 1:NPopulation)

  for (d in 1:NArea)

    for (g in 1:NGender)

    {

      a <- 1

      MeanLengthSpawning[p,d,g,a]  <-  SpawningZFraction[p,d]*MeanLengthStartOfYear[p,d,g,(a+1)]

    }

# Use MaxPlusGroup age adjustment for the plus group

for (p in 1:NPopulation)

  for (d in 1:NArea)

    for (g in 1:NGender)

    {

      tmp <- 0.0

      for (k in 0:(MaxPlusGroupAge-NAge-1))

        tmp <- tmp + exp(-k*NaturalMortality[p,d,g,NAge])*Growth.MVB((NAge+k),Length.Amin[p,d,g],Length.Amax[p,d,g],SpawningZFraction[p,d],Length.Lmin[p,d,g],Length.Lmax[p,d,g],Length.c[p,d,g])

      tmp <- tmp*(1.0-exp(-NaturalMortality[p,d,g,NAge]))

      MeanLengthSpawning[p,d,g,NAge] <- tmp/(1.0-exp(-NaturalMortality[p,d,g,NAge]*(MaxPlusGroupAge-NAge)))

    }



# Compute mean length of fishery catch at age by gender, fleet, and population

#----------------------------------------------------------------------------------------------

MeanLengthCatch <- array(rep(0.0,DimPopulationFleetGenderAge),c(NPopulation,NFleet,NGender,NAge))



for (p in 1:NPopulation)

  for (v in 1:NFleet)

    for (g in 1:NGender)

      for (a in 2:(NAge-1)) {
        age <- a-1
        MeanLengthCatch[p,FleetArea[v],g,a]  <- Growth.MVB(age,Length.Amin[p,FleetArea[v],g],Length.Amax[p,FleetArea[v],g],CatchZFraction[p,FleetArea[v]],Length.Lmin[p,FleetArea[v],g],Length.Lmax[p,FleetArea[v],g],Length.c[p,FleetArea[v],g])
      }
# Use linear interpolation for age-0 fish (index a=1)
for (p in 1:NPopulation)
  for (v in 1:NFleet)
    for (g in 1:NGender)
    {
      a <- 1
      MeanLengthCatch[p,FleetArea[v],g,a]  <-  CatchZFraction[p,FleetArea[v]]*MeanLengthStartOfYear[p,FleetArea[v],g,(a+1)]
    }
# Use MaxPlusGroup age adjustment for the plus group
for (p in 1:NPopulation)
  for (d in 1:NArea)
    for (g in 1:NGender)
    {
      tmp <- 0.0
      for (k in 0:(MaxPlusGroupAge-NAge-1))
        tmp <- tmp + exp(-k*NaturalMortality[p,d,g,NAge])*Growth.MVB((NAge+k),Length.Amin[p,d,g],Length.Amax[p,d,g],CatchZFraction[p,d],Length.Lmin[p,d,g],Length.Lmax[p,d,g],Length.c[p,d,g])
      tmp <- tmp*(1.0-exp(-NaturalMortality[p,d,g,NAge]))
      MeanLengthCatch[p,d,g,NAge] <- tmp/(1.0-exp(-NaturalMortality[p,d,g,NAge]*(MaxPlusGroupAge-NAge)))
    }


# Compute mean length of survey catch at age by gender, survey, and population
#-----------------------------------------------------------------------------------------------
MeanLengthSurvey <- array(rep(0.0,DimPopulationSurveyGenderAge),c(NPopulation,NSurvey,NGender,NAge))


for (p in 1:NPopulation)
  for (I in 1:NSurvey)
  for (g in 1:NGender)
   for (a in 2:(NAge-1)) {
    age <- a-1
    MeanLengthSurvey[p,SurveyArea[I],g,a]  <- Growth.MVB(age,Length.Amin[p,SurveyArea[I],g],Length.Amax[p,SurveyArea[I],g],SurveyZFraction[p,SurveyArea[I]],Length.Lmin[p,SurveyArea[I],g],Length.Lmax[p,SurveyArea[I],g],Length.c[p,SurveyArea[I],g])
   }
# Use linear interpolation for age-0 fish (index a=1)
for (p in 1:NPopulation)
 for (I in 1:NSurvey)
  for (g in 1:NGender)
  {
   a <- 1
   MeanLengthSurvey[p,SurveyArea[I],g,a]  <-  SurveyZFraction[p,SurveyArea[I]]*MeanLengthStartOfYear[p,SurveyArea[I],g,(a+1)]
  }
# Use MaxPlusGroup age adjustment for the plus group
for (p in 1:NPopulation)
 for (d in 1:NArea)
  for (g in 1:NGender)
  {
   tmp <- 0.0
   for (k in 0:(MaxPlusGroupAge-NAge-1))
    tmp <- tmp + exp(-k*NaturalMortality[p,d,g,NAge])*Growth.MVB((NAge+k),Length.Amin[p,d,g],Length.Amax[p,d,g],SurveyZFraction[p,d],Length.Lmin[p,d,g],Length.Lmax[p,d,g],Length.c[p,d,g])
   tmp <- tmp*(1.0-exp(-NaturalMortality[p,d,g,NAge]))
   MeanLengthSurvey[p,d,g,NAge] <- tmp/(1.0-exp(-NaturalMortality[p,d,g,NAge]*(MaxPlusGroupAge-NAge)))
  }


# Compute mean weights at age at start of year, spawning, catch,
# and survey by gender, area, and population
#-----------------------------------------------------------------------------------------------
MeanWeightStartOfYear <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))
```

```
MeanWeightSpawning <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))
MeanWeightCatch <- array(rep(0.0,DimPopulationFleetGenderAge),c(NPopulation,NFleet,NGender,NAge))
MeanWeightSurvey <- array(rep(0.0,DimPopulationSurveyGenderAge),c(NPopulation,NSurvey,NGender,NAge))


for (p in 1:NPopulation)
  for (d in 1:NArea)
    for (g in 1:NGender)
      for (a in 1:NAge) {
        MeanWeightStartOfYear[p,d,g,a]  <- WeightAtLength.Allometric(MeanLengthStartOfYear[p,d,g,a],WeightAtLength.A[p,d,g],WeightAtLength.B[p,d,g])
        MeanWeightSpawning[p,d,g,a] <- WeightAtLength.Allometric(MeanLengthSpawning[p,d,g,a],WeightAtLength.A[p,d,g],WeightAtLength.B[p,d,g])
      }
for (p in 1:NPopulation)
  for (v in 1:NFleet)
    for (g in 1:NGender)
      for (a in 1:NAge) {
        MeanWeightCatch[p,FleetArea[v],g,a] <- WeightAtLength.Allometric(MeanLengthCatch[p,FleetArea[v],g,a],WeightAtLength.A[p,FleetArea[v],g],WeightAtLength.B[p,FleetArea[v],g])
      }
for (p in 1:NPopulation)
  for (I in 1:NSurvey)
    for (g in 1:NGender)
      for (a in 1:NAge)
        MeanWeightSurvey[p,SurveyArea[I],g,a] <- WeightAtLength.Allometric(MeanLengthSurvey[p,SurveyArea[I],g,a],WeightAtLength.A[p,SurveyArea[I],g],WeightAtLength.B[p,SurveyArea[I],g])


# Store unfished equilibrium mean lengths and mean weights
#-----------------------------------------------------------------------------------------------
UnfishedMeanLengthStartOfYear <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))
UnfishedMeanLengthSpawning <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))
UnfishedMeanLengthCatch <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))
UnfishedMeanLengthSurvey <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))
```

```
UnfishedMeanWeightStartOfYear <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))

UnfishedMeanWeightSpawning <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))

UnfishedMeanWeightCatch <- array(rep(0.0,DimPopulationFleetGenderAge),c(NPopulation,NFleet,NGender,NAge))

UnfishedMeanWeightSurvey <- array(rep(0.0,DimPopulationSurveyGenderAge),c(NPopulation,NSurvey,NGender,NAge))


UnfishedMeanLengthStartOfYear <- MeanLengthStartOfYear

UnfishedMeanLengthSpawning <- MeanLengthSpawning

UnfishedMeanLengthCatch <- MeanLengthCatch

UnfishedMeanLengthSurvey <- MeanLengthSurvey

UnfishedMeanWeightStartOfYear <- MeanWeightStartOfYear

UnfishedMeanWeightSpawning <- MeanWeightSpawning

UnfishedMeanWeightCatch <- MeanWeightCatch

UnfishedMeanWeightSurvey <- MeanWeightSurvey


# Set stock-recruitment submodel parameters by gender, area, and population

#-----------------------------------------------------------------------------------------------

Recruitment.LogUnfishedR <- log(Recruitment.UnfishedR)


################################################################################################

# (1.4) INITIALIZE OBSERVATION CLASS

################################################################################################


# Set fishery selectivity submodel parameters by gender, area (fleet), and population

#-----------------------------------------------------------------------------------------------

FisherySelectivityAtAge <- array(rep(1.0,DimPopulationFleetGenderAge),c(NPopulation,NFleet,NGender,NAge))


# Compute equilibrium fishery selectivity at age by gender, area (fleet), and population

#-----------------------------------------------------------------------------------------------

EquilibriumFisherySelectivityAtAge <- array(rep(1.0,DimPopulationFleetGenderAge),c(NPopulation,NFleet,NGender,NAge))


for (p in 1:NPopulation)

  for (v in 1:NFleet)
```

```
    for (g in 1:NGender) {
      tmp <- 0.0
      for (a in 1:NAge) {
        age <- a-1
        EquilibriumFisherySelectivityAtAge[p,FleetArea[v],g,a]  <- Selectivity.Logistic(age,FisherySelectivity.a50[p,FleetArea[v],g],FisherySelectivity.slope[p,FleetArea[v],g])
      }
# Rescale to set maximum selectivity at age to be 1
      tmp <- max(EquilibriumFisherySelectivityAtAge[p,FleetArea[v],g,])
      for (a in 1:NAge) {
        EquilibriumFisherySelectivityAtAge[p,FleetArea[v],g,a] <- EquilibriumFisherySelectivityAtAge[p,FleetArea[v],g,a]/tmp
        }
    }

# Set fishery selectivity during assessment time horizon to equal equilibrium fishery selectivity
FisherySelectivityAtAge <- EquilibriumFisherySelectivityAtAge

# Set survey selectivity submodel parameters by gender, area, and population
#------------------------------------------------------------------------------------------------
SurveySelectivityAtAge <- array(rep(1.0,DimPopulationSurveyGenderAge),c(NPopulation,NSurvey,NGender,NAge))

# Compute survey selectivity at age by gender, area, and population
#------------------------------------------------------------------------------------------------
for (p in 1:NPopulation)
  for (I in 1:NSurvey)
    for (g in 1:NGender) {
      tmp <- 0.0
      for (a in 1:NAge) {
        age <- a-1
        SurveySelectivityAtAge[p,SurveyArea[I],g,a]  <- Selectivity.Logistic(age,SurveySelectivity.a50[p,SurveyArea[I],g],SurveySelectivity.slope[p,SurveyArea[I],g])
      }
    # Rescale to set maximum selectivity at age to be 1
    tmp <- max(SurveySelectivityAtAge[p,SurveyArea[I],g,])
    for (a in 1:NAge) {
      SurveySelectivityAtAge[p,SurveyArea[I],g,a] <- SurveySelectivityAtAge[p,SurveyArea[I],g,a]/tmp
      }
    }

################################################################################################
# (1.5) INITIALIZE ENVIRONMENT CLASS
################################################################################################
# Placeholder
#-----------------------------------------------------------------------------------------------

################################################################################################
# WRITE INPUT DATA TO FILE
################################################################################################

Write_Input_Data(OutputFile)

################################################################################################
#
# (2) POPULATION LOOP OVER TIME
#
################################################################################################
#
# (2.1) CALCULATE UNFISHED EQUILIBRIUM
#       POPULATION NUMBERS AND SPAWNING BIOMASS BY POPULATION, AREA, AND GENDER
#
################################################################################################
#
# Dimension arrays
#-----------------------------------------------------------------------------------------------
```

```
 EquilibriumNumbersAtAge <- array(rep(0.0,DimIterationPopulationAreaGenderAge),c(MaxIteration,NPopulation,NArea,NGender,NAge))

 EquilibriumSpawningBiomass <- array(rep(0.0,DimIterationPopulationAreaGender),c(MaxIteration,NPopulation,NArea,NGender))


# Calculate unfished equilibrium spawning biomasses

#-----------------------------------------------------------------------


source(paste(source.folder,'Calculate_Unfished_Equilibrium.R',sep=""))


########################################################################

# (2.1) END

########################################################################


########################################################################

#

# (2.2) CALCULATE FISHED EQUILIBRIUM

#       POPULATION NUMBERS AND SPAWNING BIOMASS BY POPULATION, AREA, AND GENDER

#

########################################################################


# Dimension arrays

#-------------------------------------------------

EquilibriumNumbersAtAge <- array(rep(0.0,DimIterationPopulationAreaGenderAge),c(MaxIteration,NPopulation,NArea,NGender,NAge))

EquilibriumSpawningBiomass <- array(rep(0.0,DimIterationPopulationAreaGender),c(MaxIteration,NPopulation,NArea,NGender))

EquilibriumFishingMortalityAtAge <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))

EquilibriumTotalMortalityAtAge <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))

EquilibriumRecruitmentProduction <- array(rep(0.0,DimPopulationArea),c(NPopulation,NArea))


# Calculate equilibrium fishing mortality at age

#-------------------------------------------------

for (p in 1:NPopulation)

  for (d in 1:NArea)

    for (g in 1:NGender)
```

```
   for (a in 1:(NAge)) {

    EquilibriumFishingMortalityAtAge[p,d,g,a] <- EquilibriumFisherySelectivityAtAge[p,d,g,a]*EquilibriumFishingMortality[d,d]

    EquilibriumTotalMortalityAtAge[p,d,g,a] <- NaturalMortality[p,k,g,a]+EquilibriumFishingMortalityAtAge[p,k,g,a]

   }


# Calculate fished equilibrium lengths and weights at age

# for the plus group using the MaxPlusGroupAge adjustment

#-----------------------------------------------------------------

# Use MaxPlusGroup age adjustment for the MeanLengthStartOfYear

for (p in 1:NPopulation)

 for (d in 1:NArea)

  for (g in 1:NGender)

  {

   tmp <- 0.0

   for (k in 0:(MaxPlusGroupAge-NAge-1))

    tmp <- tmp + exp(-k*EquilibriumTotalMortalityAtAge[p,d,g,NAge])*Growth.MVB((NAge+k),Length.Amin[p,d,g],Length.Amax[p,d,g],StartZFraction[p,d],Length.Lmin[p,d,g],Length.Lmax[p,d,g],Length.c[p,d,g])

   tmp <- tmp*(1.0-exp(-EquilibriumTotalMortalityAtAge[p,d,g,NAge]))

   MeanLengthStartOfYear[p,d,g,NAge] <- tmp/(1.0-exp(-EquilibriumTotalMortalityAtAge[p,d,g,NAge]*(MaxPlusGroupAge-NAge)))

  }

# Use MaxPlusGroup age adjustment for the MeanLengthSpawning

for (p in 1:NPopulation)

 for (d in 1:NArea)

  for (g in 1:NGender)

  {

   tmp <- 0.0

   for (k in 0:(MaxPlusGroupAge-NAge-1))

    tmp <- tmp + exp(-k*EquilibriumTotalMortalityAtAge[p,d,g,NAge])*Growth.MVB((NAge+k),Length.Amin[p,d,g],Length.Amax[p,d,g],SpawningZFraction[p,d],Length.Lmin[p,d,g],Length.Lmax[p,d,g],Length.c[p,d,g])

   tmp <- tmp*(1.0-exp(-EquilibriumTotalMortalityAtAge[p,d,g,NAge]))

   MeanLengthSpawning[p,d,g,NAge] <- tmp/(1.0-exp(-EquilibriumTotalMortalityAtAge[p,d,g,NAge]*(MaxPlusGroupAge-NAge)))
```

```
    }
# Use MaxPlusGroup age adjustment for the MeanLengthCatch
for (p in 1:NPopulation)
  for (d in 1:NArea)
    for (g in 1:NGender)
    {
      tmp <- 0.0
      for (k in 0:(MaxPlusGroupAge-NAge-1))
        tmp <- tmp + exp(-k*EquilibriumTotalMortalityAtAge[p,d,g,NAge])*Growth.MVB((NAge+k),Length.Amin[p,d,g],Length.Amax[p,d,g],CatchZFraction[p,d],Length.Lmin[p,d,g],Length.Lmax[p,d,g],Length.c[p,d,g])
      tmp <- tmp*(1.0-exp(-EquilibriumTotalMortalityAtAge[p,d,g,NAge]))
      MeanLengthCatch[p,d,g,NAge] <- tmp/(1.0-exp(-EquilibriumTotalMortalityAtAge[p,d,g,NAge]*(MaxPlusGroupAge-NAge)))
    }


# Use MaxPlusGroup age adjustment for the MeanLengthSurvey
for (p in 1:NPopulation)
  for (d in 1:NArea)
    for (g in 1:NGender)
    {
      tmp <- 0.0
      for (k in 0:(MaxPlusGroupAge-NAge-1))
        tmp <- tmp + exp(-k*EquilibriumTotalMortalityAtAge[p,d,g,NAge])*Growth.MVB((NAge+k),Length.Amin[p,d,g],Length.Amax[p,d,g],SurveyZFraction[p,d],Length.Lmin[p,d,g],Length.Lmax[p,d,g],Length.c[p,d,g])
      tmp <- tmp*(1.0-exp(-EquilibriumTotalMortalityAtAge[p,d,g,NAge]))
      MeanLengthSurvey[p,d,g,NAge] <- tmp/(1.0-exp(-EquilibriumTotalMortalityAtAge[p,d,g,NAge]*(MaxPlusGroupAge-NAge)))
    }


# Compute fished equilibrium mean weights of the plus group for the start of year,
# spawning, catch and survey by gender, area and population
#-----------------------------------------------------------------------------------------------
for (p in 1:NPopulation)
```

```
  for (d in 1:NArea)
    for (g in 1:NGender) {
        MeanWeightStartOfYear[p,d,g,NAge]  <- WeightAtLength.Allometric(MeanLengthStartOfYear[p,d,g,NAge],WeightAtLength.A[p,d,g],WeightAtLength.B[p,d,g])
        MeanWeightSpawning[p,d,g,NAge] <- WeightAtLength.Allometric(MeanLengthSpawning[p,d,g,NAge],WeightAtLength.A[p,d,g],WeightAtLength.B[p,d,g])
      }
for (p in 1:NPopulation)
  for (v in 1:NFleet)
    for (g in 1:NGender) {
        MeanWeightCatch[p,FleetArea[v],g,NAge] <- WeightAtLength.Allometric(MeanLengthCatch[p,FleetArea[v],g,NAge],WeightAtLength.A[p,FleetArea[v],g],WeightAtLength.B[p,FleetArea[v],g])
      }
for (p in 1:NPopulation)
  for (I in 1:NSurvey)
    for (g in 1:NGender)
        MeanWeightSurvey[p,SurveyArea[I],g,NAge] <- WeightAtLength.Allometric(MeanLengthSurvey[p,SurveyArea[I],g,NAge],WeightAtLength.A[p,SurveyArea[I],g],WeightAtLength.B[p,SurveyArea[I],g])


# Calculate fished equilibrium numbers at age
#-----------------------------------------------------------------
source(paste(source.folder,'Calculate_Fished_Equilibrium.R',sep=""))


##################################################################################################
#
# (2.3) CALCULATE ASSESSMENT TIME PERIOD
#       POPULATION NUMBERS, SPAWNING BIOMASS, AND RECRUITMENT BY POPULATION, AREA, AND GENDER
#
##################################################################################################
# Dimension Arrays
#-----------------------------------------------------------------------------------------------
NumbersAtAge <- array(rep(0.0,DimYearPlusOnePopulationAreaGenderAge),c(NYearPlusOne,NPopulation,NArea,NGender,NAge))
AssessmentMeanLengthStartOfYear <- array(rep(0.0,DimYearPopulationAreaGenderAge),c(NYear,NPopulation,NArea,NGender,NAge))
```

```
AssessmentMeanLengthSpawning <- array(rep(0.0,DimYearPopulationAreaGenderAge),c(NYear,NPopulation,NArea,NGender,NAge))

AssessmentMeanLengthCatch <- array(rep(0.0,DimYearPopulationAreaGenderAge),c(NYear,NPopulation,NArea,NGender,NAge))

AssessmentMeanLengthSurvey <- array(rep(0.0,DimYearPopulationAreaGenderAge),c(NYear,NPopulation,NArea,NGender,NAge))

AssessmentMeanWeightStartOfYear <- array(rep(0.0,DimYearPopulationAreaGenderAge),c(NYear,NPopulation,NArea,NGender,NAge))

AssessmentMeanWeightSpawning <- array(rep(0.0,DimYearPopulationAreaGenderAge),c(NYear,NPopulation,NArea,NGender,NAge))

AssessmentMeanWeightCatch <- array(rep(0.0,DimYearPopulationAreaGenderAge),c(NYear,NPopulation,NArea,NGender,NAge))

AssessmentMeanWeightSurvey <- array(rep(0.0,DimYearPopulationAreaGenderAge),c(NYear,NPopulation,NArea,NGender,NAge))


FisheryCatchNumbersAtAgeByPopulation <- array(rep(0.0,DimYearPopulationFleetAreaGenderAge),c(NYear,NPopulation,NFleet,NArea,NGender,NAge))

FisheryCatchProportionAtAgeByPopulation <- array(rep(0.0,DimYearPopulationFleetAreaGenderAge),c(NYear,NPopulation,NFleet,NArea,NGender,NAge))

FisheryCatchNumbersAtAge <- array(rep(0.0,DimYearFleetAreaGenderAge),c(NYear,NFleet,NArea,NGender,NAge))

FisheryCatchProportionAtAge <- array(rep(0.0,DimYearFleetAreaGenderAge),c(NYear,NFleet,NArea,NGender,NAge))

FisheryCatchBiomassAtAgeByPopulation <- array(rep(0.0,DimYearPopulationFleetAreaGenderAge),c(NYear,NPopulation,NFleet,NArea,NGender,NAge))

FisheryCatchBiomass <- array(rep(0.0,DimYearFleetArea),c(NYear,NFleet,NArea))


TotalMortality  <- array(rep(0.0,DimYearPopulationAreaGenderAge),c(NYear,NPopulation,NArea,NGender,NAge))


SpawningNumbersAtAge <- array(rep(0.0,DimPopulationAreaGenderAge),c(NPopulation,NArea,NGender,NAge))

SpawningBiomass <- array(rep(0.0,DimYearPopulationAreaGender),c(NYear,NPopulation,NArea,NGender))


RecruitmentProduction <- array(rep(0.0,DimYearPopulationArea),c(NYear,NPopulation,NArea))

Recruitment <- array(rep(0.0,DimYearPopulationAreaGender),c(NYear,NPopulation,NArea,NGender))


SurveyCatchability <- array(rep(0.00001,DimPopulationSurvey),c(NPopulation,NSurvey))

SurveyCatchNumbersAtAgeByPopulation <- array(rep(0.0,DimYearPopulationSurveyGenderAge),c(NYear,NPopulation,NSurvey,NGender,NAge))

SurveyCatchProportionAtAgeByPopulation <- array(rep(0.0,DimYearPopulationSurveyGenderAge),c(NYear,NPopulation,NSurvey,NGender,NAge))

SurveyCatchNumbersAtAge <- array(rep(0.0,DimYearSurveyGenderAge),c(NYear,NSurvey,NGender,NAge))

SurveyCatchProportionAtAge <- array(rep(0.0,DimYearSurveyGenderAge),c(NYear,NSurvey,NGender,NAge))

SurveyCatchBiomassAtAgeByPopulation <- array(rep(0.0,DimYearPopulationSurveyGenderAge),c(NYear,NPopulation,NSurvey,NGender,NAge))
```

```
SurveyCatchBiomass <- array(rep(0.0,DimYearSurvey),c(NYear,NSurvey))


# Calculate numbers at age and QOI during assessment period

#------------------------------------------------------------

source(paste(source.folder,'Calculate_Assessment_Period.R',sep=""))


####################################################################################################

#

# (3)	WRITE MODEL RESULTS TO OUTPUT FILE

#

####################################################################################################


# Output numbers at age and QOI during assessment period

#------------------------------------------------------------

source(paste(source.folder,'Write_Output_Data.R',sep=""))


####################################################################################################

# END

####################################################################################################
```

Appendix 2. Text input file to run the R script for MAS Case Study 1 from the current working directory.

```
# VerboseEquilibriumOutput flag (Verbose=1)
0
# NPopulation, the number of populations
2
# NAge, the number of age groups in each population
10
# MaxPlusGroupAge, the maximum age in the plus group for mean length and weight calculation, MaxPlusGroupAge>=NAge
20
# NGender, the number of genders in each population
2
# NArea, the number of areas in the model
2
# NFleet, the number of fleets in the model
2
# FleetArea, the fishing area for each fleet
1 2
# NSurvey, the number of surveys
2
# SurveyArea, the survey area for each survey
1 2
# NYear, the number of years in the assessment time horizon
20
# NSeason, the number of seasons
1
# MaxIteration, the maximum number of iterations to compute unfished and fished equilibrium
100
# EquilibriumConvergenceCriterion, the convergence criterion for equilibrium spawning biomasses
0.0001
# OutputNumbersUnits, the units for output population numbers (numbers of fish)
```

```
1000
# OutputBiomassUnits, the units for output population numbers (kg of fish)
1000
# StartZFraction, the array for timing of start of year by population and area
0 0
0 0
# SpawningZFraction, the array for timing of spawning season by population and area
0.25 0.25
0.25 0.25
# CatchZFraction, the array for timing of catch by population and area
0.5 0.5
0.5 0.5
# SurveyZFraction, the array for timing of survey by population and area
0.75 0.75
0.75 0.75
# MovementProbability, the 3D array for fish movement probability submodel parameters by population, area and area
0.9 0.75
0.1 0.25
0.1 0.25
0.9 0.75
# RecruitmentDistribution, the 3D array for recruitment movement probability submodel parameters by population, area and area
0.9 0.75
0.1 0.25
0.1 0.25
0.9 0.75
# NaturalMortality, the 4D array for natural mortality at age submodel parameters by population, area, gender and age
0.5 0.55 0.5 0.55 0.5 0.55 0.5 0.55
0.3 0.35 0.3 0.35 0.3 0.35 0.3 0.35
0.2 0.25 0.2 0.25 0.2 0.25 0.2 0.25
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
```

```
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
# Maturity.a50, the 3D array for maturity a50 parameters by population, area and gender
3.0 3.0 3.0 3.0
3.0 3.0 3.0 3.0
# Maturity.slope, the 3D array for maturity slope parameters by population, area and gender
0.1 0.5 0.1 0.5
0.1 0.5 0.1 0.5
# Length.Amin, the 3D array for length Amin parameters by population, area and gender
1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0
# Length.Amax, the 3D array for length Amax parameters by population, area and gender
7.0 7.0 7.0 7.0
7.0 7.0 7.0 7.0
# Length.Lmin, the 3D array for length Lmin parameters by population, area and gender
20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0
# Length.Lmax, the 3D array for length Lmax parameters by population, area and gender
70.0 65.0 70.0 65.0
70.0 65.0 70.0 65.0
# Length.c, the 3D array for length c parameters by population, area and gender
0.5 0.5 0.5 0.5
0.5 0.5 0.5 0.5
# WeightAtLength.A, the weight at length A parameters by population, area and gender
2.5e-5 2.5e-5 2.5e-5 2.5e-5
2.5e-5 2.5e-5 2.5e-5 2.5e-5
# WeightAtLength.B, the weight at length B parameters by population, area and gender
3.00 2.95 3.00 2.95
3.00 2.95 3.00 2.95
# Recruitment.SpawningBiomassUnits, the stock-recruitment biomass units parameters by population and area
```

```
1000.0 1000.0

1000.0 1000.0

# Recruitment.GenderFraction, the stock-recruitment gender fraction parameters by population, area and gender

0.5 0.5 0.5 0.5

0.5 0.5 0.5 0.5

# Recruitment.h, the stock-recruitment steepness parameters by population and area

0.75 0.70

0.75 0.80

# Recruitment.UnfishedR, the stock-recruitment unfished R parameters by population and area

1.0e7 1.0e7

5.0e6 1.0e7

# Recruitment.sigmaR, the stock-recruitment sigmaR parameters by population and area

0.5 0.5

0.5 0.5

# EquilibriumFishingMortality, the equilibrium fishing mortality submodel parameters by fleet and area

0.1 0.0

0.0 0.1

# FishingMortality, the fishing mortality parameters by year, fleet and area

0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2

# FisherySelectivity.a50, the fishery selectivity a50 parameters by population, fleet and gender

3.0 3.0 3.0 3.0

3.0 3.0 3.0 3.0

# FisherySelectivity.slope, the fishery selectivity a50 parameters by population, fleet and gender

0.1 0.1 0.5 0.5

0.1 0.1 0.5 0.5

# SurveySelectivity.a50, the survey selectivity a50 parameters by population, survey and gender

3.0 2.75 3.0 2.75

3.0 2.75 3.0 2.75

# SurveySelectivity.slope, the survey selectivity slope parameters by population, survey and gender

0.25 0.5 0.25 0.5

0.25 0.5 0.25 0.5
```

Appendix 3. R scripts required to be in the subdirectory named “code” to run the R script for MAS Case Study 1 (case_study_v1_20180316.R) from the current working directory. The R scripts are:

1. Read_Input_Data.R
2. Write_Input_Data.R
3. Maturity_Logistic.R
4. Growth_MVB.R
5. WeightAtLength_Allometric.R
6. Recruitment_BH.R
7. Selectivity_Logistic.R
8. Write_Output_Data.R

```
##################################################################################################

# The R program file Read_Input_Data.R reads the input data and

# results of initialization calculations for case study v1

# of the MAS project to standard input file='case_study_v1_input.txt'.

# Jon Brodziak, PIFSC, jon.brodziak@noaa.gov 13-Mar-2018

##################################################################################################

# InputFile <- 'case_study_v1_input.txt'

# OutputFile <- 'case_study_v1_output.txt'


# Read input and output file names from console

#--------------------------------------------------------------------------------------------------

print("Enter the input file name (with no spaces):")

InputFile <- scan(what="",quiet=T)

print("Input file name is:")

print(InputFile)

print("Enter the output file name (with no spaces):")

OutputFile <- scan(what="",quiet=T)

print("Output file name is:")

print(OutputFile)

SkipLines <- 1


print('________________________________________________________________________________________________________')
print('Reading input data from input file ...')
print('_____________________________________________________________________________________')

# Read the verbose output to console flag
#-----------------------------------------------------------------------------------------------
VerboseEquilibriumOutput <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Verbose console output (1=True)")
print(VerboseEquilibriumOutput)
SkipLines <- SkipLines+2

# Read the number of populations
#-----------------------------------------------------------------------------------------------
NPopulation <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Number of populations")
print(NPopulation)
SkipLines <- SkipLines+2

# Read the number of age classes including the plus group (age-0 to age-(NAge-1))
#-----------------------------------------------------------------------------------------------
NAge <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Number of ages")
print(NAge)
SkipLines <- SkipLines+2

# Read the maximum age in the plus group
#-----------------------------------------------------------------------------------------------
MaxPlusGroupAge <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Maximum plus group age")
print(MaxPlusGroupAge)
SkipLines <- SkipLines+2

# Read the number of genders
```

```
#-----------------------------------------------------------------------------------------------
NGender <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Number of genders")
print(NGender)
SkipLines <- SkipLines+2

# Read the number of areas
#-----------------------------------------------------------------------------------------------
NArea <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Number of areas")
print(NArea)
SkipLines <- SkipLines+2

# Read the number of fleets
#-----------------------------------------------------------------------------------------------
NFleet <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Number of fleets")
print(NFleet)
SkipLines <- SkipLines+2

# Read the fishing area for each fleet (1 area per fleet)
#-----------------------------------------------------------------------------------------------
FleetArea <- rep(0,length=NFleet)
FleetArea <- scan(InputFile, skip=SkipLines, n=2, quiet=T)
for (v in 1:NFleet) {
  print(c("Area for fleet",v))
  print(FleetArea[v])
}
SkipLines <- SkipLines+2

# Read the number of surveys
#-----------------------------------------------------------------------------------------------
```

```
NSurvey <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Number of surveys")
print(NSurvey)
SkipLines <- SkipLines+2

# Read the sampling area for each survey (1 area per survey)
#-----------------------------------------------------------------------------------------------
SurveyArea <- rep(0,length=NSurvey)
SurveyArea <- scan(InputFile, skip=SkipLines, n=2, quiet=T)
for (v in 1:NSurvey) {
  print(c("Area for survey",v))
  print(SurveyArea[v])
}
SkipLines <- SkipLines+2

# Read the number of years in the assessment time horizon
#-----------------------------------------------------------------------------------------------
NYear <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Number of years")
print(NYear)
SkipLines <- SkipLines+2

# Read the number of seasons in a year
#-----------------------------------------------------------------------------------------------
NSeason <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Number of seasons")
print(NSeason)
SkipLines <- SkipLines+2

# Read the maximum number of iterations to compute unfished and fished equilibrium
#-----------------------------------------------------------------------------------------------
MaxIteration <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
```

```
print("Maximum number of iterations for equilibrium calculations")
print(MaxIteration)
SkipLines <- SkipLines+2

# Set convergence criterion for calculating equilibrium spawning biomasses
#---------------------------------------------------------------------------------------------
EquilibriumConvergenceCriterion <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Convergence criterion for equilibrium calculations")
print(EquilibriumConvergenceCriterion)
SkipLines <- SkipLines+2

# Read the output units for population numbers (numbers of fish)
#---------------------------------------------------------------------------------------------
OutputNumbersUnits <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Population numbers units for output")
print(OutputNumbersUnits)
SkipLines <- SkipLines+2

# Read the output units for population biomass (kilograms of fish)
#---------------------------------------------------------------------------------------------
OutputBiomassUnits <- scan(InputFile, skip=SkipLines, n=1, quiet=T)
print("Population biomass units for output")
print(OutputBiomassUnits)
SkipLines <- SkipLines+2

# Set array dimensions
#---------------------------------------------------------------------------------------------
NYearPlusOne <- NYear+1
DimAreaGender <- NArea*NGender
DimIterationPopulationAreaGenderAge <- MaxIteration*NPopulation*NArea*NGender*NAge
DimIterationPopulationAreaGender <- MaxIteration*NPopulation*NArea*NGender
DimFleetArea <- NFleet*NArea
```

```
DimPopulationAreaGenderAge <- NPopulation*NArea*NGender*NAge
DimPopulationFleetGenderAge <- NPopulation*NFleet*NGender*NAge
DimPopulationFleetGender <- NPopulation*NFleet*NGender
DimPopulationAreaGender <- NPopulation*NArea*NGender
DimPopulationAreaAge <- NPopulation*NArea*NAge
DimPopulationAreaArea <- NPopulation*NArea*NArea
DimPopulationArea <- NPopulation*NArea
DimPopulationFleet <- NPopulation*NFleet
DimPopulationSurveyGenderAge <- NPopulation*NSurvey*NGender*NAge
DimPopulationSurveyGender <- NPopulation*NSurvey*NGender
DimPopulationSurvey <- NPopulation*NSurvey
DimSurveyGender <- NSurvey*NGender
DimYearSurvey <- NYear*NSurvey
DimYearPopulationSurveyGenderAge <- NYear*NPopulation*NSurvey*NGender*NAge
DimYearPopulationFleetAreaGenderAge <- NYear*NPopulation*NFleet*NArea*NGender*NAge
DimYearPopulationAreaGenderAge <- NYear*NPopulation*NArea*NGender*NAge
DimYearPlusOnePopulationAreaGenderAge <- NYearPlusOne*NPopulation*NArea*NGender*NAge
DimYearPopulationAreaGender <- NYear*NPopulation*NArea*NGender
DimYearSurveyGenderAge <- NYear*NSurvey*NGender*NAge
DimYearPopulationArea <- NYear*NPopulation*NArea
DimYearFleetAreaGenderAge <- NYear*NFleet*NArea*NGender*NAge
DimYearAreaGenderAge <- NYear*NArea*NGender*NAge
DimYearFleetArea <- NYear*NFleet*NArea
DimYearArea <- NYear*NArea

# Read 2D array for timing of start of year by area and population
#----------------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationArea, skip=SkipLines, quiet=T)
StartZFraction <- array(tmp,c(NPopulation,NArea))
for (p in 1:NPopulation) {
  print(c("Z fraction by area for population",p))
  print(StartZFraction[p,])
}
SkipLines <- SkipLines+NPopulation+1


# Read 2D array for timing of spawning by area and population
#---------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationArea, skip=SkipLines, quiet=T)
SpawningZFraction <- array(tmp,c(NPopulation,NArea))
for (p in 1:NPopulation) {
  print(c("Start of year Z fraction by area for population",p))
  print(SpawningZFraction[p,])
}
SkipLines <- SkipLines+NPopulation+1


# Read 2D array for timing of catch by area and population
#---------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationArea, skip=SkipLines, quiet=T)
CatchZFraction <- array(tmp,c(NPopulation,NArea))
for (p in 1:NPopulation) {
  print(c("Catch Z fraction by area for population",p))
  print(CatchZFraction[p,])
}
SkipLines <- SkipLines+NPopulation+1


# Read 2D array for timing of surveys by area and population
#---------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationArea, skip=SkipLines, quiet=T)
SurveyZFraction <- array(tmp,c(NPopulation,NArea))
for (p in 1:NPopulation) {
  print(c("Survey Z fraction by area for population",p))
  print(SurveyZFraction[p,])
}
SkipLines <- SkipLines+NPopulation+1
```

```
# Read 3D array for population movement probability parameters

#-----------------------------------------------------------------------------------------------

tmp <- scan(InputFile, nmax=DimPopulationAreaArea, skip=SkipLines, quiet=T)

MovementProbability <- array(tmp,c(NPopulation,NArea,NArea))

for (p in 1:NPopulation) {

  print(c("Movement probabilities by area for population",p))

  print(MovementProbability[p,,])

}

SkipLines <- SkipLines+DimPopulationArea+1


# Read 3D array for recruitment distribution probability parameters

#-----------------------------------------------------------------------------------------------

tmp <- scan(InputFile, nmax=DimPopulationAreaArea, skip=SkipLines, quiet=T)

RecruitmentDistribution <- array(tmp,c(NPopulation,NArea,NArea))

for (p in 1:NPopulation) {

  print(c("Recruitment Distribution by area for population",p))

  print(RecruitmentDistribution[p,,])

}

SkipLines <- SkipLines+DimPopulationArea+1


# Read 3D array for natural mortality at age parameters

#-----------------------------------------------------------------------------------------------

tmp <- scan(InputFile, nmax=DimPopulationAreaGenderAge, skip=SkipLines, quiet=T)

NaturalMortality <- array(tmp,c(NPopulation,NArea,NGender,NAge))

for (p in 1:NPopulation)

  for (d in 1:NArea)

    for(g in 1:NGender) {

      print(c("Natural mortality at age for population",p,"in area",d,"and gender",g))

      print(NaturalMortality[p,d,g,])

    }

SkipLines <- SkipLines+NAge+1
```

```
# Read 3D array for maturity a50 parameters
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationAreaGender, skip=SkipLines, quiet=T)
Maturity.a50 <- array(tmp,c(NPopulation,NArea,NGender))
for (p in 1:NPopulation)
  for (d in 1:NArea) {
    print(c("Maturity a50 parameter by gender for population",p,"in area",d))
    print(Maturity.a50[p,d,])
  }
SkipLines <- SkipLines+NGender+1

# Read 3D array for maturity slope parameters
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationAreaGender, skip=SkipLines, quiet=T)
Maturity.slope <- array(tmp,c(NPopulation,NArea,NArea))
for (p in 1:NPopulation)
  for (d in 1:NArea) {
    print(c("Maturity slope parameter by gender for population",p,"in area",d))
    print(Maturity.slope[p,d,])
  }
SkipLines <- SkipLines+NGender+1

# Read 3D array for length at age Amin parameters
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationAreaGender, skip=SkipLines, quiet=T)
Length.Amin <- array(tmp,c(NPopulation,NArea,NArea))
for (p in 1:NPopulation)
  for (d in 1:NArea) {
    print(c("Length at age Amin parameter by gender for population",p,"in area",d))
    print(Length.Amin[p,d,])
  }
```

```
SkipLines <- SkipLines+NGender+1


# Read 3D array for length at age Amax parameters
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationAreaGender, skip=SkipLines, quiet=T)
Length.Amax <- array(tmp,c(NPopulation,NArea,NArea))
for (p in 1:NPopulation)
  for (d in 1:NArea) {
    print(c("Length at age Amax parameter by gender for population",p,"in area",d))
    print(Length.Amax[p,d,])
  }
SkipLines <- SkipLines+NGender+1


# Read 3D array for length at age Lmin parameters
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationAreaGender, skip=SkipLines, quiet=T)
Length.Lmin <- array(tmp,c(NPopulation,NArea,NArea))
for (p in 1:NPopulation)
  for (d in 1:NArea) {
    print(c("Length at age Lmin parameter by gender for population",p,"in area",d))
    print(Length.Lmin[p,d,])
  }
SkipLines <- SkipLines+NGender+1


# Read 3D array for length at age Lmax parameters
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationAreaGender, skip=SkipLines, quiet=T)
Length.Lmax <- array(tmp,c(NPopulation,NArea,NArea))
for (p in 1:NPopulation)
  for (d in 1:NArea) {
    print(c("Length at age Lmax parameter by gender for population",p,"in area",d))
    print(Length.Lmax[p,d,])
  }
SkipLines <- SkipLines+NGender+1

# Read 3D array for length at age c parameters
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationAreaGender, skip=SkipLines, quiet=T)
Length.c <- array(tmp,c(NPopulation,NArea,NArea))
for (p in 1:NPopulation)
  for (d in 1:NArea) {
    print(c("Length at age c parameter by gender for population",p,"in area",d))
    print(Length.c[p,d,])
  }
SkipLines <- SkipLines+NGender+1

# Read 3D array for weight at length A parameters
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationAreaGender, skip=SkipLines, quiet=T)
WeightAtLength.A <- array(tmp,c(NPopulation,NArea,NArea))
for (p in 1:NPopulation)
  for (d in 1:NArea) {
    print(c("Weight at length at age A parameter by gender for population",p,"in area",d))
    print(WeightAtLength.A[p,d,])
  }
SkipLines <- SkipLines+NGender+1

# Read 3D array for weight at length B parameters
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationAreaGender, skip=SkipLines, quiet=T)
WeightAtLength.B <- array(tmp,c(NPopulation,NArea,NArea))
for (p in 1:NPopulation)
  for (d in 1:NArea) {
    print(c("Weight at length at age B parameter by gender for population",p,"in area",d))
    print(WeightAtLength.B[p,d,])
  }
SkipLines <- SkipLines+NGender+1

# Read 2D array for units of spawning used in recruitment models
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationArea, skip=SkipLines, quiet=T)
Recruitment.SpawningBiomassUnits <- array(tmp,c(NPopulation,NArea))
for (p in 1:NPopulation) {
  print(c("Spawning biomass units for recruitment by area for population",p))
  print(Recruitment.SpawningBiomassUnits[p,])
}
SkipLines <- SkipLines+NArea+1

# Read 3D array for gender fractions of recruitment output
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationAreaGender, skip=SkipLines, quiet=T)
Recruitment.GenderFraction <- array(tmp,c(NPopulation,NArea,NArea))
for (p in 1:NPopulation)
  for (d in 1:NArea) {
    print(c("Gender fraction for recruitment by gender for population",p,"in area",d))
    print(Recruitment.GenderFraction[p,d,])
  }
SkipLines <- SkipLines+NGender+1

# Read 2D array for steepness parameter used in recruitment models
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationArea, skip=SkipLines, quiet=T)
Recruitment.h <- array(tmp,c(NPopulation,NArea))
for (p in 1:NPopulation) {
  print(c("Steepness parameters for recruitment by area for population",p))
  print(Recruitment.h[p,])
```

```
}
SkipLines <- SkipLines+NArea+1

# Read 2D array for unfished recruitment parameter used in recruitment models
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationArea, skip=SkipLines, quiet=T)
Recruitment.UnfishedR <- array(tmp,c(NPopulation,NArea))
for (p in 1:NPopulation) {
  print(c("Unfished recruitment parameters for recruitment by area for population",p))
  print(Recruitment.UnfishedR[p,])
}
SkipLines <- SkipLines+NArea+1

# Read 2D array for sigmaR parameters used in recruitment models
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimPopulationArea, skip=SkipLines, quiet=T)
Recruitment.sigmaR <- array(tmp,c(NPopulation,NArea))
for (p in 1:NPopulation) {
  print(c("Recruitment sigmaR parameters for recruitment by area for population",p))
  print(Recruitment.sigmaR[p,])
}
SkipLines <- SkipLines+NArea+1

# Read 2D array for equilibrium fishing mortality parameters to initialize model
#-----------------------------------------------------------------------------------------------
tmp <- scan(InputFile, nmax=DimFleetArea, skip=SkipLines, quiet=T)
EquilibriumFishingMortality <- array(tmp,c(NFleet,NArea))
for (p in 1:NPopulation) {
  print(c("Equilibrium fishing mortality parameters by area for population",p))
  print(EquilibriumFishingMortality[p,])
}
SkipLines <- SkipLines+NArea+1
```

```
# Read 3D array for fishing mortality parameters in the assessment time horizon

#------------------------------------------------------------------------------------------------

tmp <- scan(InputFile, nmax=DimYearFleetArea, skip=SkipLines, quiet=T)

FishingMortality <- array(tmp,c(NYear,NFleet,NArea))

for (v in 1:NFleet)

  for (d in 1:NArea) {

    print(c("Fishing mortality parameters by year for fleet",v,"in area",d))

    print(FishingMortality[,v,d])

}

SkipLines <- SkipLines+NArea+1


# Read 3D array for fishery selectivity a50 parameters

#------------------------------------------------------------------------------------------------

tmp <- scan(InputFile, nmax=DimPopulationFleetGender, skip=SkipLines, quiet=T)

FisherySelectivity.a50 <- array(tmp,c(NPopulation,NFleet,NGender))

for (p in 1:NPopulation)

  for (v in 1:NFleet) {

    print(c("Fishery selectivity a50 parameters for population",p,"and fleet",v))

    print(FisherySelectivity.a50[p,v,])

  }

SkipLines <- SkipLines+NGender+1


# Read 3D array for fishery selectivity slope parameters

#------------------------------------------------------------------------------------------------

tmp <- scan(InputFile, nmax=DimPopulationFleetGender, skip=SkipLines, quiet=T)

FisherySelectivity.slope <- array(tmp,c(NPopulation,NFleet,NGender))

for (p in 1:NPopulation)

  for (v in 1:NFleet) {

    print(c("Fishery selectivity slope parameters for population",p,"and fleet",v))

    print(FisherySelectivity.slope[p,v,])

  }
```

```
SkipLines <- SkipLines+NGender+1


# Read 3D array for survey selectivity a50 parameters

#-------------------------------------------------------------------------------------------------

tmp <- scan(InputFile, nmax=DimPopulationSurveyGender, skip=SkipLines, quiet=T)

SurveySelectivity.a50 <- array(tmp,c(NPopulation,NSurvey,NGender))

for (p in 1:NPopulation)

  for (I in 1:NSurvey) {

    print(c("Survey selectivity a50 parameters for population",p,"and survey",I))

    print(SurveySelectivity.a50[p,I,])

  }

SkipLines <- SkipLines+NGender+1


# Read 3D array for survey selectivity slope parameters

#-------------------------------------------------------------------------------------------------

tmp <- scan(InputFile, nmax=DimPopulationSurveyGender, skip=SkipLines, quiet=T)

SurveySelectivity.slope <- array(tmp,c(NPopulation,NSurvey,NGender))

for (p in 1:NPopulation)

  for (I in 1:NSurvey) {

    print(c("Survey selectivity slope parameters for population",p,"and survey",I))

    print(SurveySelectivity.slope[p,I,])

  }

SkipLines <- SkipLines+NGender+1

###############################################################################################
```

```
###############################################################################################################
# The R program file Write_Input_Data.R writes the input data and
# results of initialization calculations for case study v1
# of the MAS project to user-named OutputFile.
# Jon Brodziak, PIFSC, jon.brodziak@noaa.gov 13-Mar-2018
###############################################################################################################
Write_Input_Data<-function(filename)
{
 print('__________________________________________________________________________________________________________')
 print('Writing input data to output file ...')
 print('__________________________________________________________________________________________________________')

 sink(filename)

 print('#################################################################################################################')
 print('# INITIIALIZING ANALYSIS CLASS')
 print('#################################################################################################################')
 print('#')

 print('#################################################################################################################')
 print('# SETTING MODEL DOMAIN PARAMETERS')
 print('#################################################################################################################')

 print('# Number of populations (1:NPopulation)')
 print(NPopulation)

 print('# Number of age classes (1:NAge)')
 print(NAge)

 print('# Maximum plus group age')
 print(MaxPlusGroupAge)
 print('# Number of genders (1:NGender)')
 print(NGender)

 print('# Number of areas (1:NArea)')
 print(NArea)

 print('# Number of fleets (1:NFleet)')
 print(NFleet)

 for (I in 1:NFleet)
 print(c("# Fleet =",I,"operates in Area=",FleetArea[I]))

 print('# Number of surveys (1:NSurvey)')
 print(NSurvey)

 for (I in 1:NSurvey)
 print(c("# Survey=",I,"operates in Area=",SurveyArea[I]))

 print('# Number of years in assessment time horizon (1:NYear)')
 print(NYear)

 print('# Number of seasons in a year (1:NSeason)')
 print(NSeason)

 print('# Maximum number of iterations for computing initial equilibrium (1:MaxIteration)')
 print(MaxIteration)

 print('# Convergence criterion for computing initial equilibrium: EquilibriumConvergenceCriterion')
 print(EquilibriumConvergenceCriterion)
 print('#')

 print('# Population numbers units for output (numbers of fish)')
 print(OutputNumbersUnits)

 print('# Population biomass units for output (kilograms of fish)')
 print(OutputBiomassUnits)

 print('####################################################################################################')
 print('# INITIIALIZING POPULATION CLASS')
 print('####################################################################################################')

 print('# Timing of start of year within year: StartZFraction[NPopulation,NArea]')
 print(StartZFraction)

 print('# Timing of spawning within year: SpawningZFraction[NPopulation,NArea]')
 print(SpawningZFraction)

 print('# Timing of catch within year: CatchZFraction[NPopulation,NFleet]')
 print(CatchZFraction)

 print('# Timing of survey within year: SurveyZFraction[NPopulation,NSurvey]')
 print(SurveyZFraction)

 print('# Population movement probability arrays: MovementProbability[NPopulation,NArea,NArea]')
 for (i in 1:NPopulation) {
   print('# Movement array for population:')
   print(i)
   print(MovementProbability[i,,])
 }

 print('# Population recruitment distribution arrays: RecruitmentDistribution[NPopulation,NArea,NArea]')
 for (i in 1:NPopulation) {
   print('# Recruitment distribution array for population:')
   print(i)
   print(RecruitmentDistribution[i,,])
 }

 print('# Population natural mortality arrays: NaturalMortality[NPopulation,NArea,NGender,NAge]')
 for (i in 1:NPopulation) {
   print('# Natural mortality at age array for population:')
   print(i)
   for (j in 1:NArea)
     for (k in 1:NGender)
       print(NaturalMortality[i,j,k,])
 }

 print('# Population maturity parameters: Maturity.a50[NPopulation,NArea,NGender]')
 for (i in 1:NPopulation) {
   print('# Age at 50% maturity array for population:')
   print(i)
   for (j in 1:NArea)
       print(Maturity.a50[i,j,])
 }

 print('# Population maturity parameters: Maturity.slope[NPopulation,NArea,NGender]')
 for (i in 1:NPopulation) {
   print('# Maturity slope parameter array for population:')
   print(i)
   for (j in 1:NArea)
     print(Maturity.slope[i,j,])
 }

 print('# Population maturity probability at age arrays: MaturityProbabilityAtAge[NPopulation,NArea,NGender,NAge]')
 for (i in 1:NPopulation) {
   print('# Maturity proabability at age array for population:')
   print(i)
   for (j in 1:NArea)
     for (k in 1:NGender)
       print(MaturityProbabilityAtAge[i,j,k,])
 }

 print('# Population growth parameters: Minimum reference age Length.Amin[NPopulation,NArea,NGender]:')
 for (i in 1:NPopulation) {
   print('# Age at amin (yr) array for population:')
   print(i)
   for (j in 1:NArea)
     print(Length.Amin[i,j,])
 }

 print('# Population growth parameters: Maximum reference age Length.Amax[NPopulation,NArea,NGender]:')
 for (i in 1:NPopulation) {
   print('# Age at amax (yr) array for population:')
   print(i)
   for (j in 1:NArea)
     print(Length.Amax[i,j,])
 }

 print('# Population growth parameters: Maximum reference age amax:')
 print('# Population growth parameters: Length.Lmin[NPopulation,NArea,NGender]')
 for (i in 1:NPopulation) {
   print('# Length at age amin (cm) array for population:')
   print(i)
   for (j in 1:NArea)
     print(Length.Lmin[i,j,])
 }

 print('# Population growth parameters: Length.Lmax[NPopulation,NArea,NGender]')
 for (i in 1:NPopulation) {
   print('# Length at age amax (cm) array for population:')
   print(i)
   for (j in 1:NArea)
     print(Length.Lmax[i,j,])
 }

 print('# Population growth parameters: Length.c[NPopulation,NArea,NGender]')
 for (i in 1:NPopulation) {
   print('# Length at age curvature parameter array for population:')
   print(i)
   for (j in 1:NArea)
     print(Length.c[i,j,])
 }

 print('# Unfished population mean length at age on January 1st arrays: MeanLengthStartOfYear[NPopulation,NArea,NGender,NAge]')
 for (i in 1:NPopulation) {
   print('# Mean length at age (cm) on January 1st array for population:')
   print(i)
   for (j in 1:NArea)
     for (k in 1:NGender)
       print(MeanLengthStartOfYear[i,j,k,])
 }

 print('# Unfished population mean length at age during spawning season arrays: MeanLengthSpawning[NPopulation,NArea,NGender,NAge]')
 for (i in 1:NPopulation) {
   print('# Mean length at age (cm) during spawning season array for population:')
   print(i)
   for (j in 1:NArea)
     for (k in 1:NGender)
       print(MeanLengthSpawning[i,j,k,])
 }
```

```
print('# Unfished population mean length at age of fishery catch arrays: MeanLengthCatch[NPopulation,Nfleet,NGender,NAge]')

for (i in 1:NPopulation) {

  print('# Mean length at age (cm) of fishery catch array for population:')

  print(i)

  for (j in 1:NFleet)

    for (k in 1:NGender)

      print(MeanLengthCatch[i,j,k,])

}


print('# Unfished population mean length at age during survey arrays: MeanLengthSurvey[NPopulation,NSurvey,NGender,NAge]')

for (i in 1:NPopulation) {

  print('# Mean length at age (cm) during survey array for population:')

  print(i)

  for (j in 1:NSurvey)

    for (k in 1:NGender)

      print(MeanLengthSurvey[i,j,k,])

}


print('# Population length-weight parameters: WeightAtLength.A[NPopulation,NArea,NGender]')

for (i in 1:NPopulation) {

  print('# Length-weight scalar parameter A array for population:')

  print(i)

  for (j in 1:NArea)

    print(WeightAtLength.A[i,j,])

}


print('# Population length-weight parameters: WeightAtLength.B[NPopulation,NArea,NGender]')

for (i in 1:NPopulation) {

  print('# Length-weight exponent parameter B array for population:')

  print(i)

  for (j in 1:NArea)

    print(WeightAtLength.B[i,j,])
```

```
}


print('# Unfished population mean weight at age on January 1st arrays: MeanWeightStartOfYear[NPopulation,NArea,NGender,NAge]')
for (i in 1:NPopulation) {
  print('# Mean weight at age (kg) on January 1st array for population:')
  print(i)
  for (j in 1:NArea)
    for (k in 1:NGender)
      print(MeanWeightStartOfYear[i,j,k,])
}


print('# Unfished population mean weight at age during spawning season arrays: MeanWeightSpawning[NPopulation,NArea,NGender,NAge]')
for (i in 1:NPopulation) {
  print('# Mean weight at age (kg) during spawning season array for population:')
  print(i)
  for (j in 1:NArea)
    for (k in 1:NGender)
      print(MeanWeightSpawning[i,j,k,])
}


print('# Unfished population mean weight at age of fishery catch arrays: MeanWeightCatch[NPopulation,NFleet,NGender,NAge]')
for (i in 1:NPopulation) {
  print('# Mean weight at age (kg) of fishery catch array for population:')
  print(i)
  for (j in 1:NFleet)
    for (k in 1:NGender)
      print(MeanWeightCatch[i,j,k,])
}


print('# Unfished population mean weight at age during survey arrays: MeanWeightSurvey[NPopulation,NSurvey,NGender,NAge]')
for (i in 1:NPopulation) {
  print('# Mean weight at age (kg) during survey array for population:')
  print(i)
  for (j in 1:NSurvey)
    for (k in 1:NGender)
      print(MeanWeightSurvey[i,j,k,])
}

print('# Units of female spawning biomass (kg) for recruitment process models: Recruitment.SpawningBiomassUnits[NPopulation,NArea]')
for (i in 1:NPopulation) {
  print('# Units of female spawning biomass (kg) by area array for population:')
  print(i)
  print(Recruitment.SpawningBiomassUnits[i,])
}

print('# Fraction of recruits by gender parameters for recruitment process models: Recruitment.GenderFraction[NPopulation,NArea]')
for (i in 1:NPopulation) {
  print('# Fraction of recruits by gender array for population:')
  print(i)
  for (j in 1:NArea)
    print(Recruitment.GenderFraction[i,j,])
}

print('# Steepness parameter h for recruitment process models: Recruitment.h[NPopulation,NArea]')
for (i in 1:NPopulation) {
  print('# Steepness parameter h by area array for population:')
  print(i)
  print(Recruitment.h[i,])
}

print('# Unfished recruitment parameter for recruitment process models: Recruitment.UnfishedR[NPopulation,NArea]')
for (i in 1:NPopulation) {
print('# Unfished recruitment parameter by area array for population:')
print(i)
print(Recruitment.UnfishedR[i,])
}


print('# Logarithm of unfished recruitment parameter for recruitment process models: Recruitment.LogUnfishedR[NPopulation,NArea]')
for (i in 1:NPopulation) {
 print('# Logarithm of unfished recruitment parameter by area array for population:')
 print(i)
 print(Recruitment.LogUnfishedR[i,])
}


print('# Standard deviation of log-scale R for recruitment process models: Recruitment.sigma.R[NPopulation,NArea]')
for (i in 1:NPopulation) {
 print('# Standard deviation of log-scale R parameter by area array for population:')
 print(i)
 print(Recruitment.sigmaR[i,])
}
print('#')


print('####################################################################################################')
print('# INITIIALIZING OBSERVATION CLASS')
print('####################################################################################################')


print('# Equilibrium fishing mortality by fleet: EquilibriumFishingMortality[NFleet,NArea]')
for (i in 1:NFleet) {
 print('# Equilibrium fishing mortality by area array for fleet:')
 print(i)
 print(EquilibriumFishingMortality[i,])
}


print('# Fishing mortality by year and fleet: FishingMortality[NYear,NFleet,NArea]')
for (i in 1:NFleet) {
 print('# Fishing mortality by year and area arrays for fleet:')
```

```
  print(i)

  for (j in 1:NArea)

    print(FishingMortality[,i,j])

}


print('# Fishery selectivity parameters for populations by fleet/area: FisherySelectivity.a50[NPopulation,NFleet,NGender]')

for (i in 1:NPopulation) {

  print('# Fishery age at 50% selectivity by fleet/area and gender arrays for population:')

  print(i)

  for (j in 1:NFleet)

    print(FisherySelectivity.a50[i,j,])

}


print('# Fishery selectivity parameters for populations by fleet/area: FisherySelectivity.slope[NPopulation,Nfleet,NGender]')

for (i in 1:NPopulation) {

  print('# Fishery selectivity slope parameter by area and gender arrays for population:')

  print(i)

  print(FisherySelectivity.slope[i,,])

}


print('# Population fishery selectivity at age arrays: FisherySelectivityAtAge[NPopulation,NFleet,NGender,NAge]')

for (i in 1:NPopulation) {

  print('# Initial fishery selectivity at age by area and gender arrays for population:')

  print(i)

  for (j in 1:NFleet)

    for (k in 1:NGender)

      print(FisherySelectivityAtAge[i,j,k,])

}


print('# Survey selectivity parameters for populations by fleet/area: SurveySelectivity.a50[NPopulation,NSurvey,NGender]')

for (i in 1:NPopulation) {

  print('# Survey age at 50% selectivity by fleet/area and gender arrays for population:')
```

```
    print(i)
    for (j in 1:NSurvey)
      print(SurveySelectivity.a50[i,j,])
  }

  print('# Survey selectivity parameters for populations by fleet/area: SurveySelectivity.slope[NPopulation,NSurvey,NGender]')
  for (i in 1:NPopulation) {
    print('# Survey selectivity slope parameter by area and gender arrays for population:')
    print(i)
    print(SurveySelectivity.slope[i,,])
  }

  print('# Population survey selectivity at age arrays: SurveySelectivityAtAge[NPopulation,NSurvey,NGender,NAge]')
  for (i in 1:NPopulation) {
    print('# Initial survey selectivity at age by area and gender arrays for population:')
    print(i)
    for (j in 1:NSurvey)
      for (k in 1:NGender)
        print(SurveySelectivityAtAge[i,j,k,])
  }
  print('#')

  print('####################################################################################################')
  print('# INITIIALIZING ENVIRONMENT CLASS')
  print('####################################################################################################')
  print('#')

  sink()
  return()
}
####################################################################################################
```

```
##################################################################################################################
Maturity.Logistic <- function(age,a50,slope) {
  fraction.mature <- 1.0/(1.0+exp(-(age-a50)/slope))
  return(fraction.mature)
}
##################################################################################################################
Growth.MVB <- function(age,amin,amax,offset,Lmin,Lmax,c) {
  tmp <- 0.0
  tmp <- (1.0-c^(age+offset-amin))/(1.0-c^(amax-amin))
  length  <- Lmin+tmp*(Lmax-Lmin)
  return(length)
}
##################################################################################################################
WeightAtLength.Allometric <- function(length,A,B) {
  weight <- A*(length^B)
  return(weight)
}
##################################################################################################################
Recruitment.BH <- function(SB, SB0, R0, h) {
  R <- 4*h*R0*SB/(SB0*(1-h)+SB*(5*h-1))
  return(R)
}
##################################################################################################################
Selectivity.Logistic <- function(age,a50,slope) {
  selectivity <- 1.0/(1.0+exp(-(age-a50)/slope))
  return(selectivity)
}
##################################################################################################################
```

```
####################################################################################################
# The R script Write_Output_Data.R writes the output data and
# results of assessment period calculations for case study v1
# of the MAS project to user-named OutputFile.
# Jon Brodziak, PIFSC, jon.brodziak@noaa.gov 13-Mar-2018
####################################################################################################
print('________________________________________________________________________________________________________')
print('Writing output data to output file ...')
print('________________________________________________________________________________________________________')

sink(file=OutputFile,append=TRUE,type="output")


print('________________________________________________________________________________________________________')
print('CASE STUDY 1 RESULTS')
print('________________________________________________________________________________________________________')
#--------------------------------------------------------------------------------------------------
# (3.1.1)  FISHERY SELECTIVITY
#--------------------------------------------------------------------------------------------------
print('________________________________________________________________________________________________________')
print('(3.1.1) FISHERY SELECTIVITY (Population Fleet Gender(1=Females)) AT AGE')
print('________________________________________________________________________________________________________')
for (p in 1:NPopulation)
  for (v in 1:NFleet)
    for (g in 1:NGender)
    {
      print(c("Population=",p,"Fleet=",v,"Gender=",g,"By Age",FisherySelectivityAtAge[p,FleetArea[v],g,]))
    }
print('________________________________________________________________________________________________________')


#--------------------------------------------------------------------------------------------------
# (3.1.2)  SURVEY SELECTIVITY
#--------------------------------------------------------------------------------------------------
```

```
print('(3.1.2) SURVEY SELECTIVITY (Population Survey Gender(1=Females)) AT AGE')

print('____________________________________________________________________________________________')

for (p in 1:NPopulation)

  for (I in 1:NSurvey)

    for (g in 1:NGender)

    {

      print(c("Population=",p,"Survey=",I,"Gender=",g,"By Age",SurveySelectivityAtAge[p,SurveyArea[I],g,]))

    }

print('____________________________________________________________________________________________')


#-------------------------------------------------------------------------------------------------

# (3.1.3)  FISHING MORTALITY

#-------------------------------------------------------------------------------------------------

print('(3.1.3) FISHING MORTALITY (Year Fleet Area_A Area_B)')

print('____________________________________________________________________________________________')

for (v in 1:NFleet)

  for (y in 1:NYear)

  {

    print(c("Year=",y,"Fleet=",v,"By Area",FishingMortality[y,FleetArea[v],]))

  }

print('____________________________________________________________________________________________')


#-------------------------------------------------------------------------------------------------

# (3.1.4)  TOTAL MORTALITY

#-------------------------------------------------------------------------------------------------

print('(3.1.4) TOTAL MORTALITY (Year Population Area/Fleet Gender(1=Females) AT AGE)')

print('____________________________________________________________________________________________')

for (p in 1:NPopulation)

  for (d in 1:NArea)

    for (g in 1:NGender)

      for (y in 1:NYear)

      {
        print(c("Year=",y,"Population=",p,"Area=",d,"Gender=",g,"By Age",TotalMortality[y,p,d,g,]))
      }
print('____________________________________________________________________________________________________')

#-----------------------------------------------------------------------------------------------
# (3.1.5)  FEMALE SPAWNING BIOMASS
#-----------------------------------------------------------------------------------------------
print('(3.1.5) FEMALE SPAWNING BIOMASS (Population Area/Fleet Year')
print('____________________________________________________________________________________________________')
for (p in 1:NPopulation)
  for (d in 1:NArea)
    for (y in 1:NYear)
    {
      print(c("Population=",p,"Area=",d,"Year=",y,SpawningBiomass[y,p,d,1]*Recruitment.SpawningBiomassUnits/OutputBiomassUnits))
    }
print('____________________________________________________________________________________________________')

#-----------------------------------------------------------------------------------------------
# (3.1.6)  RECRUITMENT
#-----------------------------------------------------------------------------------------------
print('(3.1.6) RECRUITMENT (Population Area/Fleet Year')
print('____________________________________________________________________________________________________')
for (p in 1:NPopulation)
  for (d in 1:NArea)
    for (y in 1:NYear)
    {
      print(c("Population=",p,"Area=",d,"Year=",y,(Recruitment[y,p,d,1]+Recruitment[y,p,d,2]/OutputNumbersUnits)))
    }
print('____________________________________________________________________________________________________')

#-----------------------------------------------------------------------------------------------
# (3.1.7)  TOTAL MORTALITY
```

```
#----------------------------------------------------------------------------------------------------

print('(3.1.7) TOTAL MORTALITY (Year Population Area/Fleet Gender(1=Females) AT AGE)')

print('____________________________________________________________________________________________________')

for (p in 1:NPopulation)

  for (d in 1:NArea)

    for (g in 1:NGender)

      for (y in 1:NYear)

      {

        print(c("Year=",y,"Population=",p,"Area=",d,"Gender=",g,"By Age",TotalMortality[y,p,d,g,]))

      }

print('____________________________________________________________________________________________________')




#----------------------------------------------------------------------------------------------------

# (3.1.8)  FISHERY OBSERVATIONS

#----------------------------------------------------------------------------------------------------

print('(3.1.8) FISHERY CATCH NUMBERS AT AGE (Year Population Fleet Gender(1=Females))')

print('____________________________________________________________________________________________________')

for (p in 1:NPopulation)

  for (v in 1:NFleet)

    for (d in 1:NArea)

      if (FleetArea[v]==d)

        for (g in 1:NGender)

          for (y in 1:NYear)

          {

            print(c("Year=",y,"Population=",p,"Fleet=",v,"Gender=",g,"By
Age",FisheryCatchNumbersAtAgeByPopulation[y,p,FleetArea[v],d,g,]/OutputNumbersUnits))

          }

print('____________________________________________________________________________________________________')

print('(3.1.8) FISHERY CATCH PROPORTION AT AGE (Year Population Area/Fleet Gender(1=Females))')

print('____________________________________________________________________________________________________')

for (p in 1:NPopulation)
```

```
  for (v in 1:NFleet)
    for (d in 1:NArea)
      if (FleetArea[v]==d)
        for (g in 1:NGender)
          for (y in 1:NYear)
          {
            print(c("Year=",y,"Population=",p,"Fleet=",v,"Gender=",g,"By Age",FisheryCatchProportionAtAgeByPopulation[y,p,FleetArea[v],d,g,]))
          }
print('____________________________________________________________________________________________________')
print('(3.1.8) FISHERY CATCH BIOMASS AT AGE (Year Population Area/Fleet Gender(1=Females))')
print('____________________________________________________________________________________________________')
for (p in 1:NPopulation)
  for (v in 1:NFleet)
    for (d in 1:NArea)
      if (FleetArea[v]==d)
        for (g in 1:NGender)
          for (y in 1:NYear)
          {
            print(c("Year=",y,"Population=",p,"Fleet=",v,"Gender=",g,"By Age",FisheryCatchBiomassAtAgeByPopulation[y,p,FleetArea[v],d,g,]/OutputBiomassUnits))
          }
print('____________________________________________________________________________________________________')



#-----------------------------------------------------------------------------------------------
# (3.1.9)  SURVEY OBSERVATIONS
#-----------------------------------------------------------------------------------------------
print('(3.1.9) SURVEY CATCH NUMBERS AT AGE (Year Population Area Gender(1=Females))')
print('____________________________________________________________________________________________________')
for (p in 1:NPopulation)
  for (I in 1:NSurvey)
    for (g in 1:NGender)
```

```
      for (y in 1:NYear)

      {

        print(c("Year=",y,"Population=",p,"Survey=",I,"Gender=",g,"By Age",SurveyCatchNumbersAtAgeByPopulation[y,p,SurveyArea[I],g,]))

      }

print('________________________________________________________________________________________________')

print('(3.1.9) SURVEY CATCH PROPORTION AT AGE (Year Population Area Gender(1=Females))')

print('________________________________________________________________________________________________')

for (p in 1:NPopulation)

  for (I in 1:NSurvey)

    for (g in 1:NGender)

      for (y in 1:NYear)

      {

        print(c("Year=",y,"Population=",p,"Survey=",I,"Gender=",g,"By Age",SurveyCatchProportionAtAgeByPopulation[y,p,SurveyArea[I],g,]))

      }

print('________________________________________________________________________________________________')

print('(3.1.9) SURVEY CATCH BIOMASS AT AGE (Year Population Area Gender(1=Females))')

print('________________________________________________________________________________________________')

for (p in 1:NPopulation)

  for (I in 1:NSurvey)

    for (g in 1:NGender)

      for (y in 1:NYear)

      {

        print(c("Year=",y,"Population=",p,"Survey=",I,"Gender=",g,"By Age",SurveyCatchBiomassAtAgeByPopulation[y,p,SurveyArea[I],g,]))

      }

print('________________________________________________________________________________________________')


#-------------------------------------------------------------------------------------------

# (3.1.10)  QUANTITIES OF INTEREST

#-------------------------------------------------------------------------------------------

print('(3.1.10) FISHERY CATCH NUMBERS AT AGE (Year Area/Fleet Gender(1=Females))')

print('________________________________________________________________________________________________')

for (v in 1:NFleet)
```

```
 for (d in 1:NArea)
  if (FleetArea[v]==d)
   for (g in 1:NGender)
    for (y in 1:NYear)
    {
     print(c("Year=",y,"Fleet=",v,"Gender=",g,"By Age",FisheryCatchNumbersAtAge[y,FleetArea[v],d,g,]/OutputNumbersUnits))
    }
print('__________________________________________________________________________________________')
print('(3.1.10) FISHERY CATCH PROPORTION AT AGE (Year Fleet Gender(1=Females))')
print('__________________________________________________________________________________________')
print('__________________________________________________________________________________________')
for (v in 1:NFleet)
 for (d in 1:NArea)
  if (FleetArea[v]==d)
   for (g in 1:NGender)
    for (y in 1:NYear)
    {
     print(c("Year=",y,"Area/Fleet=",v,"Gender=",g,"By Age",FisheryCatchProportionAtAge[y,FleetArea[v],d,g,]))
    }
print('__________________________________________________________________________________________')
print('(3.1.10) FISHERY CATCH BIOMASS (Year Area/Fleet)')
print('__________________________________________________________________________________________')
for (v in 1:NFleet)
 if (FleetArea[v]==d)
  for (d in 1:NArea)
   for (y in 1:NYear)
   {
    print(c("Year=",y,"Fleet=",v,FisheryCatchBiomass[y,FleetArea[v],d]/OutputBiomassUnits))
   }
print('__________________________________________________________________________________________')
print('(3.1.10) SURVEY CATCH NUMBERS AT AGE (Year Area Gender(1=Females))')
print('__________________________________________________________________________________________')
```

```
for (I in 1:NSurvey)
  for (g in 1:NGender)
    for (y in 1:NYear)
    {
      print(c("Year=",y,"Survey=",I,"Gender=",g,"By Age",SurveyCatchNumbersAtAge[y,SurveyArea[I],g,]))
    }
print('____________________________________________________________________________________________________')
print('(3.1.10) SURVEY CATCH PROPORTION AT AGE (Year Area Gender(1=Females))')
print('____________________________________________________________________________________________________')
for (I in 1:NSurvey)
  for (g in 1:NGender)
    for (y in 1:NYear)
    {
      print(c("Year=",y,"Survey=",I,"Gender=",g,"By Age",SurveyCatchProportionAtAge[y,SurveyArea[I],g,]))
    }
print('____________________________________________________________________________________________________')
print('(3.1.10) SURVEY CATCH BIOMASS (Year Area)')
print('____________________________________________________________________________________________________')
for (I in 1:NSurvey)
  for (y in 1:NYear)
  {
    print(c("Year=",y,"Survey=",I,SurveyCatchBiomass[y,SurveyArea[I]]))
  }
print('____________________________________________________________________________________________________')


#---------------------------------------------------------------------------------------------
# (3.1.11)  POPULATION NUMBERS AT AGE ON JANUARY 1ST BY YEAR
#---------------------------------------------------------------------------------------------
print('(3.1.11) POPULATION NUMBERS AT AGE  ON JANUARY 1ST BY YEAR (Year Population Area Gender(1=Females))')
print('____________________________________________________________________________________________________')
for (p in 1:NPopulation)
  for (d in 1:NArea)
```

```
    for (g in 1:NGender)
      for (y in 1:NYear)
      {
        print(c("Year=",y,"Population=",p,"Area=",d,"Gender=",g,"By Age",NumbersAtAge[y,p,d,g,]/OutputNumbersUnits))
      }
print('____________________________________________________________________________________________')


#-----------------------------------------------------------------------------------------------
# (3.1.12)  POPULATION MEAN LENGTHS AT AGE ON JANUARY 1ST BY YEAR
#-----------------------------------------------------------------------------------------------
print('(3.1.12) POPULATION MEAN LENGTHS AT AGE ON JANUARY 1ST BY YEAR (Year Population Area Gender(1=Females))')
print('____________________________________________________________________________________________')
for (p in 1:NPopulation)
  for (d in 1:NArea)
    for (g in 1:NGender)
      for (y in 1:NYear)
      {
        print(c("Year=",y,"Population=",p,"Area=",d,"Gender=",g,"By Age",AssessmentMeanLengthStartOfYear[y,p,d,g,]))
      }
print('____________________________________________________________________________________________')

#-----------------------------------------------------------------------------------------------
# (3.1.13)  POPULATION MEAN WEIGHTS AT AGE ON JANUARY 1ST BY YEAR
#-----------------------------------------------------------------------------------------------
print('(3.1.12) POPULATION MEAN WEIGHTS AT AGE ON JANUARY 1ST BY YEAR (Year Population Area Gender(1=Females))')
print('____________________________________________________________________________________________')
for (p in 1:NPopulation)
  for (d in 1:NArea)
    for (g in 1:NGender)
      for (y in 1:NYear)
      {
```

```
     print(c("Year=",y,"Population=",p,"Area=",d,"Gender=",g,"By Age",AssessmentMeanWeightStartOfYear[y,p,d,g,]))
    }
print('________________________________________________________________________________')


sink()


print('________________________________________________________________________________')
print('Calculations are complete for MAS Case Study 1')
print('________________________________________________________________________________')
```