User Manual for Stock Synthesis - Model Version 3.24s

Richard D. Methot Jr.

SEDAR39-RD-13

October 2014

User Manual for
Stock Synthesis

Model Version 3.24s

Updated November 21, 2013

Richard D. Methot Jr.
NOAA Fisheries
Seattle, WA

Table of Contents

1. Introduction

This manual provides a guide for using the stock assessment program, Stock Synthesis (SS). The guide contains a description of the input and output files and usage instructions. A technical description of the model itself is in Methot and Wetzel (2013). SS is programmed using Auto Differentiation Model Builder (ADMB; Fournier 2001. ADMB is now available at admb-project.org). SS currently is compiled using ADMB version 11.1 using the Microsoft C++ Optimizing Compiler Version 16.0. The model and a graphical user interface are available from the NOAA Fisheries Stock Assessment Toolbox website: http://nft.nefsc.noaa.gov/. An output processor package, r4ss, in R is available for download from CRAN and additional information about the package can be located at http://code.google.com/p/r4ss/.

2. File Organization

2.1 Input Files

1. *STARTER.SS*: required file containing filenames of the data file and the control file plus other run controls (required).
2. <*datafile*>: file containing model dimensions and the data with file extension .dat (required)
3. <*control file*>: file containing set-up for the parameters with file extension .ctl (required)
4. *FORECAST.SS*: file containing specifications for forecasts (required)
5. *SS3.PAR*: previously created parameter file that can be read to overwrite the initial parameter values in the control file (optional)
6. *RUNNUMBER*.SS: file containing a single number used as runnumber in output to CUMREPORT.SSO and in the processing of PROFILEVALUES.SS (optional)
7. *PROFILEVALUES*.SS: file contain special conditions for batch file processing (optional)

2.2 Output Files

1. *SS3.PAR, SS3.STD, SS3.REP, SS3.COR* etc. standard ADMB output files
2. *Echoinput.sso*: This file is produced while reading the input files and includes an annotated echo of the input. The sole purpose of this output file is debugging input errors.
3. *Warning*.sso: This file contains a list of warnings generated during program execution.
4. *checkup.sso*: Contains details of selectivity parameters and resulting vectors. This is written during the first call of the objective function.
5. *Report*.sso: This file is the primary report file.
6. *CompReport*.sso: Beginning with version 3.03, the composition data has been separated into a dedicated report
7. *FORECAST-REPORT.sso*: Output of management quantities and for forecasts
8. *cumreport.sso:* This file contains a brief version of the run output, output is appended to current content of file so results of several runs can be collected together. This is useful when a batch of runs is being processed.
9. *Covar.sso:* This file replaces the standard ADMB ss3.cor with an output of the parameter and derived quantity correlations in database format

10. *data.ss_new*: contains a user-specified number of datafiles, generated through a parametric bootstrap procedure, and written sequentially to this file
11. *Control.ss_new*: Updated version of the control file with final parameter values replacing the Init parameter values.
12. *Starter.ss_new*: New version of the starter file with annotations
13. *Forecast.ss_new*: New version of the forecast file with annotations.
14. *REBUILD.DAT*: Output formatted for direct input to Andre Punt's rebuilding analysis package. Cumulative output is output to REBUILD.SS (useful when doing MCMC or profiles).

2.3 Auxiliary Files

1. *SS3-OUTPUT.XLS*: Excel file with macros to read *report.sso* and display results
2. *SELEX24_dbl_normal.XLS*:
    a. This excel file is used to show the shape of a double normal selectivity (option number 20 for age-based and 24 for length-based selectivity) given user-selected parameter values.
    b. Instructions are noted in the XLS file but, to summarize
        i. Users should only change entries in a yellow box.
        ii. Parameter values are changed manually or using sliders, depending on the value of cell I5.
    c. It is recommend that users select plausible starting values for double-normal selectivity options, especially when estimating all 6 parameters
    d. Please note that the XLS does NOT show the impact of setting parameters 5 or 6 to "-999". In SS3, this allows the the value of selectivity at the initial and final age or length to be determined by the shape of the double-normal arising from parameters 1-4, rather than forcing the selectivity at the intial and final age or length to be estimated separately using the value of parameters 5 and 6.
3. *SELEX17_age_randwalk.XLS*:
    a. This excel file is used to show the shape of age-based selectivity arising from option 17 given user-selected parameter values
    b. Users should only change entries in the yellow box.
    c. The red box is the maximum cumulative value, which is subtracted from all cumulative values. This is then exponentiated to yield the estimated selectivity curve. Positive values yield increasing selectivity and negative values yield decreasing selectivity.
4. *PRIOR-TESTER.XLS*:
    a. The "compare" tab of this spreadsheet shows how the various options for defining parameter priors work
5. *SS-Control_Setup.XLS*:
    a. Shows how to setup an example control file for SS
6. *SS-Data_Input.XLS*:
    a. Shows how to setup an example data input for SS
7. *Growth.XLS*:
    a. Excel file to test parameterization between the growth curve options within SS.

   b. Instructions are noted in the XLS file but, to summarize
      i. Users should only change entries in a yellow box.
      ii. Entries in a red box are used internally, and can be compared with other parameterizations, but should not be changed.
   c. The SS-VB is identical to the standard VB, but uses a parameterization where length is estimated at pre-defined ages, rather than A=0 and A=Inf. The Schnute-Richards is identical to the Richards-Maunder, but similarly uses the parameterization with length at pre-defined ages. The Richards coefficient controls curvature, and if the curvature coefficient = 1, it reverts to the standard VB curve.
8. *Movement.XLS*:
   a. Excel file to explore SS movement parameterization.

3. Starting SS

SS runs as a DOS program with text-based input. The executable is named *ss3.exe*. It can be run at the command prompt in a DOS window, or called from another program, such as R or the SS-GUI or a DOS batch file. See the section in this manual on use of batch file which can allow *ss3.exe* to reside in a separate directory. Sometimes you may receive a version of SS with array checking turned on (*SS-safe.exe*) or without array checking (*SS_opt.exe*). In this case, it is recommended to rename the one you are planning to use to *SS3.exe* before running it. Communication with the program is through text files. When the program first starts, it reads the file *STARTER.SS*, which must be located in the same directory from which SS is being run. The file *STARTER.SS* contains required input information plus references to other required input files, as described in the File Organization section. Output from SS is as text files containing specific keywords. Output processing programs, such as the SS GUI, Excel, or R can search for these keywords and parse the specific information located below that keyword in the text file.

4. Computer Requirements and Recommendations

SS is compiled to run under DOS with a 32-bit or 64-bit Windows operating system. It is recommended that the computer have at least a 2.0 Ghz processor and 2 GB of RAM. In addition SS has now been successfully compiled in Linux.

5. Starter File

SS begins by reading the file STARTER.SS. Its format and content is as follows. Note that the term COND in the Typical Value column means that the existence of input shown there is conditional on a value specified earlier in the file. Omit or comment out these entries if the appropriate condition has not been selected.

STARTER.SS		
Typical Value	**Options**	**Description**
#C this is a starter comment	Must begin with #C then rest of line is free form	All lines in this file beginning with #C will be retained and written to the top of several output files
NewFeatures.DAT	NewFeatures.DAT	Filename of the data file
NewFeatures.ctl	NewFeatures.ctl	Filename of the control file
0	Initial parameter values 0=use values in control file; 1=use ss3.par after reading setup in the control file	Don't use this if there have been any changes to the control file that would alter the number or order of parameters stored in the SS3.par file. Values in SS3.par can be edited, carefully
1	Run display detail 0=none other than ADMB outputs 1=one brief line of display for each iteration 2=fuller display per iteration	With option 2, the display shows value of each logL component for each iteration and it displays where crash penalties are created
1	Detailed age-structured report in REPORT.SSO 0 = omit catch-at-age for each fleet and cohort 1 = include all output	
0	Check-up 0=omit 1=write detailed intermediate calculations to CHECKUP.SSO during first call	This output is largely unformatted and undocumented and is mostly used by the developer.

0	Parameter Trace 0=omit 1=write good iterations and active parms 2= write good iterations and all parms; 3= write every_iter and all parms 4= write every_iter and active parms	This controls the output to PARMTRACE.SSO The contents of this output can be used to determine which values are changing when a model approaches a crash condition. It also can be used to investigate patterns of parameter changes as model convergence slowly moves along a ridge
1	Cumulative Report 0= Omit 1= Brief 2= Full	Controls reporting to the file CUMREPORT.SSO. This cumulative report is most useful when accumulating summary information from likelihood profiles or when simply accumulating a record of all model runs within the current subdirectory
1	Full Priors 0= only calculate priors for active parameters 1= calculate priors for all parameters that have a defined prior	Turning on this option causes all prior values to be calculated. With this option off, the total logL, which includes the logL for priors, would change between model phases as more parameters became active
1	Soft Bounds 0= omit 1= use	This option creates a weak symmetric beta penalty for the selectivity parameters. This becomes important when estimating selectivity functions in which the values of some parameters cause other parameters to have negligible gradients, or when bounds have been set too widely such that a parameter drifts into a region in which it has negligible gradient. The soft bound creates a weak penalty to move parameters away from the bounds.
1	Data File Output 0= none 1= Output an annotated replicate of the input data file; 2= Add a second data file containing the model's expected values with no error added 3 to N = Add N-2 parametric bootstrap data files	All output files are sequentially output to DATA.SS_new and will need to be parsed by the user into separate data files. The output of the input data file makes no changes, so retains the order of the original file Output files 2-N contain only observations that have not been excluded through use of the negative year denotation, and the order of these output observations is as processed by the model. The N obs values are adjusted accordingly. At this time, the tag recapture data is not output to DATA.SS_new

8	Turn off estimation -1: exit after reading input files 0: exit after one call to the calculation routines and production of SSO and SS_New files; <positive value>: exit after completing this phase	The "0" option is useful for (1) quickly reading in a messy set of input files and producing the annotated CONTROL.SS_new and DATA.SS_new files, or (2) examining model output based solely on input parameter values. Similarly, the value option allows examination of model output after completing a specified phase. Also see usage note for restarting from a specified phase.
10	MCMC burn interval	Need to document this and set good default
2	MCMC thin interval	Need to document this and set good default
0.0	Jitter A positive value here will add a small random jitter to the initial parameter values.	The *Jitter* factor works by adding: random normal deviate * *Jitter* * (Parm_max-Parm_min), to the initial value of the parameter. Except, (Parm_max-Parm_min) is replaced by the value 4.0 when applying *Jitter* to the recruitment deviation vector.
-1	SD report start -1: begin annual SD report in start year <year>: begin Sdreport this year	
-1	SD report end -1: end annual SD report in end year -2: end annual SD report in last forecast year <value>: end SD annual report in this year	
2	Extra SD report years 0: none <value>: number of years to read	In a long time series application, the model variance calculations will be smaller and faster if not all years are included in the SD reporting. For example, the annual SD reporting could start in 1960 and the extra option could select reporting in each decade before then.
COND: If Extra SD report years > 0		
1940 1950	Vector of years for additional SD reporting	
0.0001	Final convergence	This is a reasonable default value for the change in logL denoting convergence. For applications with much data and thus a large total logL value, a larger convergence criterion may still provide acceptable convergence
0	Retrospective year 0: none -X: retrospective year relative to end year	Adjusts the model end year and disregards data after this year. May not handle time varying parameters completely.

2	Summary biomass min age	Minimum integer age for inclusion in the summary biomass used for reporting and for calculation of total exploitation rate
1	Depletion basis 0: skip; 1: X*B0; 2: X*Bmsy; 3: X*B_styr	Selects the basis for the denominator when calculating degree of depletion in SSB. The calculated values are reported to the SD report
0.40	Fraction (X) for Depletion denominator	So would calculate the ratio SSB_y / ($0.40*SSB_0$)
1	SPR report basis 0: skip; 1: use 1-SPR_{target}; 2: use 1-SPR at MSY; 3: use 1-SPR at B_{target}; 4: no denominator, so report actual 1-SPR values	SPR is the equilibrium SSB per recruit that would result from the current year's pattern and intensity of F's. The SPR approach to measuring fishing intensity was implemented because the concept of a single annual F does not exist in SS. The quantities identified by 1, 2, and 3 here are all calculated in the benchmarks section. Then the one specified here is used as the selected denominator in a ratio with the annual value of (1.0 – SPR). This ratio (and its variance) is reported to the SD report output for the years selected above in the SD report year selection.

4	F_std report value 0: skip; 1: exploitation rate in biomass; 2: exploitation rate in numbers; 3: Sum(full F's by fleet) 4: Population F for range of ages	In addition to SPR, an additional proxy for annual F can be specified here. As with SPR, the selected quantity will be calculated annually and in the benchmarks section. The ratio of the annual value to the selected (see F report basis below) benchmark value is reported to the SD report vector. Options 1 and 2 use total catch for the year and summary abundance at the beginning of the year, so combines seasons and areas. But if most catch occurs in one area and there is little movement between areas, this ratio is not informative about the F in the area where the catch is occurring. Option 3 is a simple sum of the full F's by fleet, so may provide non-intuitive results when there are multi areas or seasons or when the selectivities by fleet do not have good overlap in age. Option 4 is a real annual F calculated as a numbers weighted F for a specified range of ages (read below). The F is calculated as Z-M where Z and M are each calculated an ln(N(t+1)/N(t)) with and without F active, respectively. The numbers are summed over all biology morphs and all areas for the beginning of the year, so subsumes any seasonal pattern.
COND: If F_reporting = 4		
13 17	Age range if F_reporting = 4	Specify range of ages. Upper age must be less than maxage because of incomplete handling of the accumulator age for this calculation.
1	F report basis 0: not relative, report raw values; 1: use F_std value corresponding to SPR_{target}; 2: use F_std value corresponding to F_{msy}; 3: use F_std value corresponding to $F_{Btarget}$	Selects the denominator to use when reporting the F_std report values. Note that order of these options differs from the biomass report basis options
999	End of file	Must be 999 in first 3 columns with no preceding tab or spaces in order to be correctly parsed by the #C comment reader

6. Forecast File

The specification of options for forecasts is contained in the mandatory input file named *FORECAST.SS*. For additional detail on the forecast file see Appendix B.

FORECAST.SS			
Typical Value		**Options**	**Description**
#C forecast comment			Same syntax and usage as in STARTER file
1		Benchmarks 0: omit; 1: calc F_spr, F_btgt, and F_msy	SS checks for consistency of the Forecast specification and the benchmark specification. It will turn benchmarks on if necessary and report a warning.
1		Forecast Method 0=none; 1: F(SPR); 2: F(MSY); 3: F(Btarget); 4: F(endyr); 5: Ave recent F (enter yrs) – not yet implemented 6: read Fmult – not yet implemented	Specifies whether or not to do a forecast and which F to use for that forecast. Basis for some additional conditional input
COND: 0 - 4		No additional input for these options	
COND: 5			
	-4	First year for recent ave F relative to end year	Read a range of years for calculation of recent average F (not yet implemented)
	0	Last year for recent ave F	Will be used to calculate an average F multiplier for each fleet over a range of years
COND: 6			
	0.6	F multiplier for option 6 (not yet implemented)	
0.45		SPR_{target}	SS searches for F multiplier that will produce this level of spawning biomass (Reproductive output) per recruit relative to unfished value.
0.40		$Biomass_{target}$	SS searches for F multiplier that will produce this level of spawning biomass relative to unfished value. This is not “per recruit” and takes into account the Spawner-Recruitment relationship

0 0 0 0 0 0	Benchmark Years >0: absolute year <=0: year relative to end year	Requires 6 years entered, beginning and ending years for biology, selectivity, and relative Fs, that will be used in to calculate benchmark quantities. Option to enter the actual year or values of 0 or negative integer values that will set the value to the model ending year. (beg_bio; end_bio; beg_selex; end_selex; beg_relF; end_relF)
1	Benchmark Relative F Basis 1 = use year range 2 = set relF same as forecast below	
2	Forecast 0: none (no forecast years) 1: set to F(SPR); 2: search for F(MSY); 3: set to F(Btgt); 4: set to F(endyr) 5: input annual F scalar	This input is ignored in benchmarks are turned off, but its existence is not conditional on benchmark switch. If Benchmarks are on, then F_spr and FBtgt are calculated. This MSY switch determines whether F_MSY is also calculated or is set to one of these other quantities
10	N forecast years	
1	F scalar	Only used if Forecast option = 5 (input annual F scalar)
0 0 0 0	Forecast Years >0: absolute year <=0: year relative to end year	Requires 4 years entered, beginning and ending years for selectivity and relative Fs that will be used in population forecasts. Option to enter the actual year or values of 0 or negative integer values that will set the value to the model ending year. (beg_selex; end_selex; beg_relF; end_relF)
1	Control Rule 1: catch = f(SSB) U.S. West Coast 2: F = f(SSB)	

0.4	Control Rule Upper Limit	Biomass level (as fraction of B0) above which F is constant
0.1	Control Rule Lower Limit	Biomass level (as fraction of B0) below which F is set to 0.0
0.75	Control Rule Buffer	Multiplier applied to forecast F before calculating catch
3	Number of Forecast Loops	Number of loops ranging from 1-3, currently fixed at 3 within SS
3	First forecast loop with stochastic recruitment	
0	Forecast loop control #3	Reserved for future model features
0	Forecast loop control #4	Reserved for future model features
0	Forecast loop control #5	Reserved for future model features
2015	First Year for Caps and Allocations	Should be after years with fixed inputs
0	Implimentation Error	The log of the ratio between the realized catch and the target catch in the forecast. (set value > 0.0 to cause active implementation error)
0	<u>Rebuilder</u> 0: omit West Coast rebuilder output 1: do rebuilder output	
2004	<u>Rebuilder start year (Y initial)</u> >0: year for current age structure; -1: set to endyear+1	
1	<u>Fleet Relative F</u> 1: use firt-last alloc year 2=read sas(row) x fleet(col) set below	
2	<u>Basis for Maximum Forecast Catch</u> 2: cap in terms of total catch biomass; 3: cap in terms of retained catch biomass; 5: cap in terms of total catch numbers; 6: cap in terms of retained catch numbers.	

COND: 2 (Conditional input for Fleet Relative F)			
	0.1 0.8 0.1	Fleet Allocation by Relatve F Fraction	The fraction of the forecast F value. For a multiple area model user must define a fraction for each fleet and each area. The total fractions must sum to one over all fleets and areas. Ex: #Fleet 1 Fleet 2 …Fleet X 0.10 0.10 0.30 # Area1 0.10 0.10 0.30 # Area2
-1 -1 -1		<u>Max Total Catch by Fleet</u> -1: no max	Must enter value for each fleet
-1		<u>Max Total Catch by Area</u> -1: no max	Must enter value for each area
0 0 0		<u>Fleet Assignment to Allocation Group</u> 0: Fleet not included in allocation group	Enter group ID# for each fleet
COND: >0			
0		Forecast Catch Levels	Number of forecast catch levels to input, else calculated from the forecast F
COND: >0			
3		<u>Basis for Forecast Catch</u> 2: Dead Catch 3: Reatined Catch 99: Input Harvest Rate(F)	
COND: >0			
	2012 1 1 1200	Year Season Fleet Catch (or F value)	
999		End of Input	

7. Optional Input Files

RUNNUMBER.SS

This file contains a single integer value. It is read when the program starts, incremented by 1, used when processing the profile value inputs (see below), used as an identifier in the batch output, then saved with the incremented value. Note that this incrementation may not occur if a run crashes.

PROFILEVALUES.SS

This file contains information for changing the value of selected parameters for each run in a batch. In the ctl file, each parameter that will be subject to modification by *PROFILEVALUES.SS* is designated by setting its phase to -9999.

The first value in *PROFILEVALUES.SS* is the number of parameters to be batched. This value MUST match the number of parameters with phase set equal to -9999 in the ctl file. The program performs no checks for this equality. If the value is zero in the first field, then nothing else will be read. Otherwise, the model will read runnumber * Nparameters values and use the last Nparameters of these to replace the initial values of parameters designated with phase = -9999 in the ctl file.

USAGE Note:

If one of the batch runs crashes before saving the updated value of runnumber.ss, then the processing of the profilevalue.ss will not proceed as expected. Check the output carefully until a more robust procedure is developed.

8. Data File

8.1 Overview of Data File

1. Dimensions (years, ages, N fleets, N surveys, etc.)
2. Fleet and survey names, timing, etc.
3. Catch biomass
4. Discard
5. Mean body weight
6. Length composition set-up
7. Length composition
8. Age composition set-up
9. Ageing imprecision definitions
10. Age composition
11. Mean length or bodyweight-at-age
12. Generalized size composition (e.g. weight frequency)
13. Tag-recapture
14. Stock composition
15. Environmental data

8.2 Units of Measure

The normal units of measure are as follows:

Catch biomass – metric tons

Body weight – kilograms

Body length – usually in cm, weight at length parameters must correspond to the units of body length and body weight.

Survey abundance – any units if q is freely scaled; metric tons or thousands of fish if q has a quantitative interpretation

Output biomass – metric tons

Numbers – thousands of fish, because catch is in mtons and body weight is in kg

Spawning biomass – metric tons of mature females if eggs/kg = 1 for all weights; otherwise has units that are proportional to egg production.

8.3 Data File Syntax

8.3.1 Model Dimensions

Typical Value	Description
#C data using new survey	Data file comment. Must start with #C to be retained then written to top of various output files. These comments can occur anywhere in the data file, but must have #C in columns 1-2.
1971	Start year
2001	End year
1	N seasons per year
12	Vector with N months in each season. These do not need to be integers
	Note: If the sum of this vector is close to 12.0, then it is rescaled to sum to 1.0 so that season duration is a fraction of a year. But if the sum is not close to 12.0, then the entered values are simply divided by 12. So with one season per year and 3 months per season, the calculated season duration will be 0.25, which allows a quarterly model to be run as if quarters are years. All rates in SS are annual rates (growth, mortality, etc.) and the realized rate for each season is the annual rate * season duration.
1	Spawning season; spawning biomass is calculated at the beginning of this season and used as basis for the total annual recruitment.
2	N fishing fleets
2	N surveys; These become numbered sequentially following the fisheries. With 2 fisheries and 2 surveys, the surveys will be fleet (aka “type”) 3 and 4.
1	Number of areas
Comm_Fish%Recr_Fish %Trawl_survey%Recruits	String containing names for each fishery and survey, delimited by the “%” character

-0.7 0.5 0.6 0.75	Sample Timing: A vector containing the timing of each fishery and survey for the sole purpose of specifying the amount of age-specific mortality that occurs before the expected values are calculated. Values are the fraction of the season elapsed before the CPUE is measured or the survey conducted. In a multiple season setup, this timing fraction is the same in each season. This fraction affects only the timing of the numbers at age calculation; the size-at-age is always calculated just at the midpoint of the season. For the fishery data, this timing based approach means that the catch-at-age sampled will be different than the actual catch-at-age. The timing approach is a straightforward exponential calculation with continuous F and a timing value can be selected that will closely approximation the catch-at-age according to the standard catch equation. For Pope's the actual catch-at-age is calculated after $e^{-M*0.5*seasdur}$ has occurred so is not at all responsive to catch that occurs within the season. The timing approach was implemented in SS so the fishery catch-at-age sample with Pope's would approximate the catch-at-age sample from continuous F. The downside is that the sample catch-at-age with Pope's diverges from the actual catch-at-age in that season. A new option is implemented with SS v3.03. For the sample timing of fishing fleets (but not surveys), a negative value for Sample Timing will cause SS to use the actual catch-at-age and not a timed sample of catch-at-age.
1 2 1 1	Vector with area assignment for each fishery and survey
1 2	Vector with units of catch for each fleet (but not entered for the surveys) 1: biomass; 2: numbers
0.05 0.10	Vector with std.err. of log(catch) for each fleet (but not entered for the surveys). These values are used only in calculating the logL for the initial equilibrium catch and for the catch deviations if the F_Method is 2 or 3. If year-specific std.err. values are needed, then use the advanced set-up in the control file. NOTE: Many users set the catch_se equal to a small value, say 0.01. During the parameter search, such high precision on catch will cause a high logL gradient when using Fmethod=2 (F as parameters). Same is probably true when estimating the init_F parameters. Suggest using a larger catch_se, say 0.05 or so.
2	Number of genders (1/2); females are gender 1
40	Accumulator age. This should be large enough so that fish at this age will be a very large (say 99%) of Linfinity. It also is important that this age be larger than the maximum age bin for the age data so that misaged old fish will principally have an assigned age that is still within the oldest age data bin. Note: SS always starts at age 0.

8.3.2 Catch

The catch data input has been modified to make it possible to maintain a long time series of catch data and then use just a subset for a particular model run. This requires explicitly reading a date field for each catch record. After reading the initial equil catch, read an integer with the number of catch input lines. Then read that number of input lines where each line has a vector of catch by fleet followed by the year and the season. Records that are before startyear or after endyr are dropped from the catch table used in the model.

In addition, it is possible to collapse the number of seasons. So if a season value is greater than the N seasons for a particular model, that catch is added to the catch for N seasons. This is generally to collapse a seasonal model into an annual model. In a seasonal model, use of season=0 will cause SS to distribute the input value of catch equally among the N seasons.

Previously, the input of catch units (biomass vs. numbers) for each fleet did not occur until the setup of the catchability characteristics for each fleet's CPUE. It was easily overlooked in that location. Now, a vector of Catch _Units for fleets (not surveys) is read right after reading season timing in the data file. Not that this is separate from the units specified in the catchability section, so it would be possible to specify a fleet's catch in biomass and CPUE in numbers.

After reading fleet-specific catch_units, read fleet-specific catch standard error. These values were implicitly set to 0.01 in SS2. In SS V3, they are used when calculating the likelihood for the initial equilibrium catch and for calculating the likelihood for catch when using F_method 2 or F_method 3. The catch stderr is meaningless when using Pope's approximation because the harvest rate is calculated to match the catch exactly.

100 1250	Initial equilibrium catch for each fishery. Enter annual values even if model has >1 season. If there are more than 1 seasons, then model will use first season body size and selectivity in calculating expected catch.		
96	Number of catch records to read below Any records that are outside the year range, start year to end year, are ignored		
Fishery 1 Catch	Fishery 2 Catch	Year	Season
10	30	1950	1
15	22	1951	1
…			

- Catch can in be terms of biomass or numbers for each fleet.
- Catch is retained catch. If there is discard data also, then it is handled in the discard section below.
- If there is reason to believe that the retained catch values underestimate the true catch, then it is possible in the retention parameter set-up to create the ability for the model to estimate the degree of unrecorded catch.

8.3.3 Abundance Indices

<table>
<tr><td colspan="2">16</td><td colspan="3">#N observations (Need to do manual count and enter N here)</td></tr>
<tr><td colspan="2">#_Fleet/Survey ID Numer</td><td>Units (0=num; 1=bio; 2=F)</td><td colspan="2">Error distribution (-1=normal; 0=lognorm; >0=degrees of freedom T-dist)</td></tr>
<tr><td colspan="2">1</td><td>1</td><td colspan="2">0</td></tr>
<tr><td colspan="2">2</td><td>1</td><td colspan="2">0</td></tr>
<tr><td colspan="2">3</td><td>0</td><td colspan="2">0</td></tr>
<tr><td>Year</td><td>Seas</td><td>Fleet/Survey</td><td>Value</td><td>Std.err. of log(value)</td></tr>
<tr><td>1991</td><td>1</td><td>3</td><td>80000</td><td>0.056</td></tr>
<tr><td>1995</td><td>1</td><td>3</td><td>65000</td><td>0.056</td></tr>
<tr><td>……….</td><td></td><td></td><td></td><td></td></tr>
<tr><td>2000</td><td>1</td><td>4</td><td>42000</td><td>0.056</td></tr>
</table>

Concept

- For fishing fleets. CPUE is defined in terms of retained catch (biomass or numbers). For fishery independent surveys, retention/discard is not defined so CPUE is implicitly in terms of total CPUE. If a survey has its selectivity mirrored to that of a fishery, only the selectivity is mirrored so the expected CPUE for this mirrored survey is in terms of total catch. Also, fishing effort is related to F, which is the F for total catch.
- If the statistical analysis used to create the CPUE index of a fishery has been conducted in such a way that its inherent size/age selectivity differs from the size/age selectivity estimated from the fishery's size and age composition, then you may want to enter the CPUE as if it was a separate survey and with a selectivity that differs from the fishery's estimated selectivity. The need for this split arises because the fishery size and age composition should be derived through a catch-weighted approach (to appropriately represent the removals by the fishery) and the CPUE should be derived through an area-weighted GLM (to appropriately serve as if it was a survey of stock abundance).
- If the fishery or survey has time-varying selectivity, then this changing selectivity will be taken into account when calculating expected values for the CPUE or survey index.

Fleet/Survey

- Fishing fleets and surveys are consecutively numbered throughout SS (e.g. three fishing fleets have ID numbers 1, 2 and 3; then 2 surveys get ID numbers 4 and 5).
- By explicitly adding the ID number to this input table, future capability to use only a subset of included fleets/surveys is being built.

Units

- NOTE: the "effort" option can only be used for a fishing fleet and not for a survey, even if the survey is mirrored to a fishing fleet. The values of these effort data are interpreted as proportional to the level of the fishery F values. No adjustment is made for differentiating between continuous F values versus exploitation rate values coming from Pope's approximation. A normal error structure is recommended so that the input effort data are compared directly to the model's calculated F, rather than to $log_e(F)$. The resultant proportionality constant has units of 1/Q.

Error distribution

- -1 = normal error
- 0 = lognormal error
- >0 = Student's T-distribution in log space with degrees of freedom equal to this value. For DF>30, results will be nearly identical to that for lognormal distribution. A DF value of about 4 gives a fat-tail to the distribution (see Chen (2003)). The se values entered in the data file must be the standard error in log_e space.
- Abundance indices typically have a lognormal error structure with units of standard error of log_e(index). If the variance of the observations is available only as a CV, then the value of se can be approximated as $\sqrt{log_e(1 + \mathrm{CV}^2)}$ where CV is the standard error of the observation divided by the mean value of the observation.
- For the normal error structure, the entered values for se are interpreted directly as a se in arithmetic space and not as a CV. Thus switching from a lognormal to a normal error structure forces the user to provide different values for the se input in the data file.
- If the data exist as a set of normalized Z-scores, you can either: assert a lognormal error structure after entering the data as exp(Z-score) because it will be logged by SS. Preferably, the Z-scores would be entered directly and the normal error structure would be used.

Data Format

- Year values that are before start year or after end year are excluded from model, so the easiest way to include provisional data in a data file is to put a negative sign on its year value.
- Duplicate survey observations are not allowed.
- Observations can be entered in any order, except if the super-year feature is used.
- Observations that are to be included in the model but not included in the –logL need to have a negative sign on their fleet ID. Previously the code for not using observations was to enter the observation itself as a negative value. However, that old approach prevented use of a Z-score environmental index as a "survey".
- Super-periods are turned on and then turned back off again by putting a negative sign on the season. Previously, super-periods were started and stopped by

entering -9999 and the -9998 in the se field. See the "Data Super-Period" section of this manual for more information.

- Special Surveys: Four special kinds of surveys are defined in SS. Here in the survey data section, there is no change in the way in which these survey data are entered. Then in the size-selectivity section of the control file, the selectivity pattern used to generate expected values for these surveys is specified by entering the selectivity pattern as 30, 31, 32, or 33. These four survey "selectivity" pattern options bypass the calculation of survey selectivity from explicit selectivity parameters.

Pattern Number	Expected Value equals:	Description
30	Spawning Biomass	Spawning biomass: e.g. for a egg&larvae survey
31	Exp(Recruitment deviation)	Useful for environmental index affecting recruitment
32	SpawnBio * Exp(RecrDev)	For a pre-recruit survey occurring before density-dependence
33	Recruitment	Age 0 recruits

8.3.4 Discard

If discard is not a feature of a model specification, then just 2 inputs are needed:

0	# N fleets with discard
0	# N discard observations

If discard is being used, the input syntax is:

3	# N fleets with discard			
#Fleet	Units	Error		
1	2	-1		
3	2	-1		
2	#N observations			
#Year	Seas	Fleet	Observation	Error
1980	1	1	.05	0.25
1991	1	1	0.10	0.25

Discard Units:
1: values are amount of discard in either biomass or numbers according to the selection made for retained catch;
2: values are fraction (in biomass or numbers) of total catch discarded; bio/num selection matches that of retained catch
3: values are in numbers (thousands) of fish discarded, even if retained catch has units of biomass

Discard Error Structure:

- The four options for DF_disc are:
  - >=1: Degrees of freedom for Student's T distribution used to scale mean body weight deviations. Value of error in data file is interpreted as CV of the observation;

    - 0: normal distribution. Value of error in data file is interpreted as CV of the observation;
    - -1: normal distribution. Value of error in data file is interpreted as standard error of the observation;
    - -2: lognormal distribution. Value of error in data file is interpreted as standard error of the observation in log space.
- Data Format
    - Year values that are before start year or after end year are excluded from model, so the easiest way to include provisional data in a data file is to put a negative sign on its year value.
    - Negative value for fleet causes it to be included in the calculation of expected values, but excluded from the logL
    - 0.0 is a legitimate discard observation, unless lognormal error structure is used
    - Duplicate survey observations are not allowed.
    - Observations can be entered in any order, except if the super-period feature is used.
- Cautionary Note: The use of CV as the measure of variance can cause a small discard value to appear to be overly precise, even with the minimum std.err. of the discard observation set to 0.001. In the control file, there is an option to add an extra amount of variance. This amount is added to the std.err., not to the CV, to help correct this problem of underestimated variance.

<u>8.3.5 Mean Body Weight</u>

This is the overall mean body weight across all selected sizes and ages. This may be useful in situations where individual fish are not measured but mean weight is obtainded by counting the number of fish in a specified sample, e.g. a 25 kg basket. Version 3.24r added the capability to use mean length data by modifying the mean weight data approach. Now observations can be entered in terms of mean length by setting switching the partition code to 10=all, 11=discard, and 12=retained rather than the 0,1, and 2 typically used with the mean body weight approach.

2	#N observations				
30	#Degrees of freedom for Student's T distribution used to evaluate mean body weight deviations. (Not conditional, must be here even if no mean body wt observations.)				
#Year	Seas	Fleet	Partition	Value	CV
1990	1	1	0	4.	0.95
1990	1	1	1	1.	0.95

- Units must correspond to the units of body weight (or mean length in cm), normally kilograms.
- Error is entered as the coefficient of variation (CV) of the observed mean bodywt (or mean length).
- Mean bodywt observations (or mean length) have a T-distribution error structure.
- Expected value of mean bodywt (or mean length) is calculated in a way that incorporates effect of selectivity and retention

- New specification that first appears here is “Partition”, where:
  - 0 means whole catch in units of weight (discard+retained)
  - 1 means discarded catch in units of weight
  - 2 means retained catch in units of weight
  - 10 means whole catch in units of length (discard+retained)
  - 11 means discarded catch in units of weight
  - 12 means retained catch in units of length

8.3.6 Population Length Bins

The beginning of the length composition section sets up the bin structure for both the population and for the length composition data.

		Length bin method. This creates a Conditional read situation below: 1=use databins; 2=generate from binwidth,min,max below 3=read vector
COND 1		
	1	Selects option 1; No additional input necessary
COND 2		
	2	Selects option 2; Read 3 additional input values
	2 10 82	Binwidth, lower size of first bin, lower size of largest bin. The number of bins is then calculated from: (maxLread-minLread)/binwidth2+1
COND 3		
	3	Selects option 3; Read 1 value, then read vector of bin boundaries
	25	number of population length bins to be read
	26 28 30 …	Vector containing lower edge of each population size bin
End of Conditional inputs for Length Bin Method		

Notes:

For option 2, binwidth should be a factor of min size and max size. For options 2 and 3, the population length bins must not be wider than the length data bins, but the boundaries of the bins do not have to align. In SS_v3.02B and earlier, the data boundaries needed to align with the population boundaries but this requirement has been removed. The transition matrix is output to *checkup.sso*.

The mean size at age 0.0 (virtual recruitment age) is set equal to the min size of the first population length bin.

Note that in SS2, the minimum size of fish in the population was defined to be equal to the smallest data size bin. If the growth curve defined any smaller fish, these were simply assigned to this first size bin. Also, size selectivity for the smallest fish was calculated on the basis of the first size bin and, implicitly, any smaller fish would have the same selectivity. Now, population size composition and size-selectivity are defined for size bins that may be smaller than the smallest size data bin. This could cause some change in the details of the fit for previously constructed SS models. The selectivity pattern #24 has been modified (see below) to reduce this effect.

When using more population length bins than data bins, SS will run slower (more calculations to do), the calculated weights at age will be less aliased by the bin structure, and you may or may not get better fits to your data.

While exploring the performance of models with finer bin structure, a potentially pathological situation has been identified. When the bin structure is coarse (note that some applications have used 10 cm bin widths for the largest fish), it is possible for a selectivity slope parameter or a retention parameter to become so steep that all of the action occurs within the range of a single size bin. In this case, the model will lose the gradient of the logL with respect to that parameter and convergence will be hampered. A generic guidance to avoid this situation is not yet available.

Length Composition

-0.0001	Compress tails of composition until observed proportion is greater than this value; negative value causes no compression; Advise using no compression if data are very sparse, and especially if the set-up is using agecomp within length bins because of the sparseness of these data
0.0001	Constant added to observed and expected proportions at length and age to make logL calculation more robust; Tail compression occurs before adding this constant Proportions are renormalized to sum to 1.0 after constant is added
0	Combine males into females at or below this bin number. This is useful if the gender determination of very small fish is doubtful so allows the small fish to be treated as combined gender. If CombGender>0, then add males into females for bins 1 thru this number, zero out the males, set male data to start at the first bin above this bin. Note that CombGender is entered as a bin index, not as the size associated with that bin. Comparable option is available for age composition data.
22	N bins in the length composition data
32 34 …	Vector containing lower edge of length bins. The last length bin will have same width as next lower bin

- Bin width does not need to be uniform for either the population (with option 3) or length composition data
- The tail compression and added constant are used in the processing of both the length composition and the age composition data. They do not apply to the generalized size composition data.
- SS will compare the population bin boundaries to the length data bin boundaries and will create a fatal error if there is any misalignment. However, for the generalize size composition data (see below), there is no need for alignment because SS will interpolate as necessary.
- If broad length bins are used, then beware of steep selectivity and retention parameters. An overly steep curve can disappear within the domain of a single length bin, thus causing ADMB to lose track of its gradient
- The mean weight-at-length, maturity-at-length and size-selectivity are based on the mid-length of the population bins. So these quantities will be rougher approximations if broad bins are defined.
- Provide a wide enough range of population size bins so that the mean body weight-at-age will be calculated correctly for the youngest and oldest fish. If the growth curve

extends beyond the largest size bin, then these fish will be assigned a length equal to the mid-bin size for the purpose of calculating their body weight.

- More bins create a bigger model internal structure and slower run times
- When fish recruit at age 0.0, they are assigned a size equal to the lower edge of the smallest population size bin.
- Fish smaller than the first data bin are placed in the first bin.

30	N Length comp observations

Example of a single length composition observation:

Year	Seas	Fleet	Gender	Partition	Nsamp	data vector
1986	1	1	3	0	20	<female then male data>

- In a 2 gender model, the data vector always has female data followed by male data, even if only one of the two genders has data that will be used (see "gender" note below).
- Each observation can be stored as one row for ease of data management in a spreadsheet and for sorting of the observations. However, the 6 header values, the female vector and the male vector could each be on a separate line because ADMB reads values consecutively from the input file and will move to the next line as necessary to read additional values.
- The composition observations can be in any order. However, if the super-period approach is used, then each super-periods' observations must be contiguous in the data file.
- Gender Flag: If model has only one gender defined in the set-up, all observations must have gender set equal to 0 or 1.
- Gender flag in 2 gender model:
  - Gender = 0 means combined male and female (must already be combined and information placed in the female portion of the data vector) (entries in male portion of vector must exist and will be ignored).
  - Gender = 1 means female only (male entries must exist for correct data reading, then will be ignored)
  - Gender = 2 means male only (female entries must exist and will be ignored after being read)
  - Gender = 3 means both data from both genders will be used and they are scaled so that they together sum to 1.0
- Partition indicates discard vs. retained (0=combined; 1=discard; 2=retained)
- If the value of year is negative, then that observation is not transferred into the working array. This feature is the easiest way to include observations in a data file but not to use them in a particular model scenario.
- If the value of fleet is negative, then the observation is processed and its logL is calculated, but this logL is not included in the total logL. This feature allows the user to see the fit to a provisional observation without having that observation affect the model.

8.3.7 Age Composition

The age composition section begins by reading a definition of the age bin structure, then the definition of ageing imprecision, then the age composition data itself. The bins are in terms of observed age (here age'). The ageing imprecision definitions are used to create one or more matrices to translate true age structure into expected age structure in terms of age'.

<table>
<tr><td>17</td><td colspan="16">N age' bins;
 can be equal to 0 if age data not used;
do not include a vector of agebins if Nage'bins is set equal to 0;</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>20</td><td>25</td></tr>
</table>

- Above is the vector with lower age of age' bins
- The first and last bins work as accumulators. So in this example any age 0 fish that are caught would be accumulated into the age'1 bin.

8.3.7.1 Ageing error

Here, the capability to create a distribution of age' (e.g. age with possible bias and imprecision) from true age is created. One or many age error definitions can be created. For each, there is input of a vector of mean age' and stddev of age'. For one definition, the input vectors can be replaced by vectors created from estimable parameters. In the future, capability to read a full age' – age matrix could be created.

2	Number of ageing error matrices to generate

- In principle, one could have year, or laboratory specific matrices
- If no age data, there can be 0 vectors
- For each matrix, enter a vector with mean age' for each true age; if there is no ageing bias, then set age' equal to true age + 0.5. Alternatively, -1 value for mean age' means to set it equal to true age plus 0.5. The addition of +0.5 is needed so that fish will get assigned to the intended interger age'
- The length of the input vector is Nage+1, with the first entry being for age 0 fish and the last for fish of age Nage.
- Followed by a vector with the stddev of age' for each true age
- The following table shows the values for the first 5 ages for each of two age transition definitions: the first defines a matrix with no bias and negligible imprecision. The second shows a small negative bias beginning at age 4.

For age 0	Age 1	Age 2	Age 3	Age 4. Etc.
-1	-1	-1	-1	-1
0.001	0.001	0.001	0.001	0.001
0.5	1.5	2.5	3.5	4.3
0.5	0.65	0.67	0.7	0.8

- SS is abile to create one ageing error matrix from parameters, rather than from an input vector. The range of conditions in which this newfeature will perform well has not been evaluated, so it should be considered as a preliminary implementation and subject to modification.

- To invoke this option, for the selected ageing error vector, set the stddev of ageing error to a negative value for age 0. This will cause creation of an ageing error matrix from parameters and any age or size-at-age data that specify use of this ageerr pattern will use this matrix.Then in the control file, add 7 parameters below the cohort growth dev parameter. These parameters are described in the control file section of this manual.

26	N age observations
1	Length bin range method for Lbin_lo and Lbin_hi: 1 = value refers to population length bin index 2 = value refers to length data bin index 3 = value is an actual length (which must correspond to a population length bin boundary)
1	Combine males into females below this age bin number. Note this is in terms of age' bins, not true age.

An example age composition observation

Year	Seas	Fleet	Gender	Partition	ageerr	Lbin lo	Lbin hi	Nsamp	Data Vector
1987	1	1	3	0	2	-1	-1	79	Enter data values

- Syntax for Gender, Partition, and data vector are same as for length
- Ageerr identifies which ageing error matrix to use to generate expected value for this observation
- The data vector has female values then male values, just as for the length composition data.
- As with the length comp data, a negative value for year causes the observation to not be read into the working matrix, a negative value for Nsamp causes the observation to be included in expected values calculation, but not in contribution to total logL.
- Lbin lo, and Lbin hi are the range of length bins that this age composition observation refers to. Normally these are entered with a value of 1 and Maxbin. Whether these are entered as population bin number, length data bin number, or actual length is controlled by the value of the Length bin range method above.
  - Entering value of 0 or –1 for Lbin lo converts Lbin lo to 1;
  - Entering value of 0 or –1 for Lbin hi converts Lbin hi to Maxbin;
  - It is strongly advised to use the "-1" codes to select the full size range. If you use explicit values, then the model could unintentionally exclude information from some size range if the population bin structure is changed.
  - In reporting to the comp_ report.sso, the reported Lbin_lo and Lbin_hi values are always converted to actual length.

8.3.7.2 Conditional Age'-at-Length:

- When Lbin_lo and Lbin_hi are used to select a subset of the total size range, the expected value for these age' data is calculated within that specified size range, so is age' conditional on length.
- In a two gender model, it is best to enter these conditional age'-at-length data as single gender observations (gender =1 for females and =2 for males), rather than as joint gender observations (gender=3). In this way, it isolates the age composition data from any gender selectivity as well.

- Use of conditional age'-at-length will greatly increase the total number of age' composition observations and associated model run time, but it is a superior approach because it:
    - Avoids double use of fish for both age' and size information because the age' information is considered conditional on the length information;
    - Contains more detailed information about the relationship between size and age so provides stronger ability to estimate growth parameters, especially the variance of size-at-age;
    - Where age data are collected in a length-stratified program, the conditional age'-at-length approach can directly match the protocols of the sampling program.

8.3.7.3 Sex Ratio-at-length

The conditional age'-at-length approach can be used to analyze sex ratio-at-length data. If you have no age data, then the following simple setup will allow entry of sex-ratio at length. Note that it must use the joint gender (code 3) approach.

1 #_N_age_bins # so all fish are put into a single "age" bin regardless of their true age
10 # assigned "age" for this one bin
1 #_N_ageerror_definitions
10.5 10.5 10.5 10.5 10.5 10.5 repeat for each true age in model, beginning at age 0
0.001 0.001 0.001 0.001 0.001 0.001 … repeat for each true age in model, beginning at age 0

1 #_N_Agecomp_obs
1 #_Lbin_method: 1=poplenbins; 2=datalenbins; 3=lengths
0 #_combine males into females at or below this bin number
There are 4 females and 8 males in the 25th population length bin

1	#_N_age_bins # so all fish are put into a single "age" bin regardless of their true age														
10	# assigned "age" for this one bin														
1	#_N_ageerror_definitions														
10.5	10.5	10.5	10.5	…	repeat for each true age in model, beginning at age 0										
0.001	0.001	0.001	0.001	…	repeat for each true age in model, beginning at age 0										
1	#_N_Agecomp_obs														
1	#_Lbin_method: 1=poplenbins; 2=datalenbins; 3=lengths														
0	#_combine males into females at or below this bin number														
# There are 4 females and 8 males in the 25th population length bin															
#Yr	Seas	Flt/Svy	Gender	Part	Ageerr	Lbin_lo	Lbin_hi	Nsamp							
1971	1	1	3	0	1	25	25	12	0	4	0	0	8	0	…

If you have both real age data and sex ratio at length data, then you will need to set up the N_age_bins to match the real age data, define an additional age_error type to use for the sex ratio data, put the sex ratio data into the correct bin. For example,

5	#_N_age_bins
1 2 3 4 5	# assigned "age" for this one bin

2		#_N_ageerror_definitions											
-1	1	1	1	1	extend for each true age in model, beginning at age 0								
0.2	0.4	0.5	0.8	1	extend for each true age in model, beginning at age 0								
3.5	3.5	3.5	3.5	3.5	repeat for each true age in model, beginning at age 0								
0.001	0.001	0.001	0.001	…	repeat for each true age in model, beginning at age 0								
2		#_N_Agecomp_obs											
1		#_Lbin_method: 1=poplenbins; 2=datalenbins; 3=lengths											
0		#_combine males into females at or below this bin number											
# There are 4 females and 8 males in the 25th population length bin													
#Yr	Seas	Flt/ Svy	Gender	Part	Ageerr	Lbin_lo	Lbin_hi	Nsamp					
1971	1	1	3	0	1	-1	-1	25	1	2	4	…	#real age data 5
1971	1	1	3	0	2	25	25	12	0	0	4	…	#sex ratio in bin 3

8.3.8 Mean Length or bodywt-at-age

SS also accepts input of mean length-at-age' or mean bodywt-at-age'. This is done in terms of age', not true age, to take into account the effects of ageing imprecision on expected mean size-at-age'. If the value of "AgeErr" is positive, then the observation is interpreted as mean length-at-age'. . If the value of "AgeErr" is negative, then the observation is interpreted as mean bodywt-at-age' and the abs(AgeErr) is used as AgeErr.

An example observation:

1	N size@age' observations									
Year	Seas	Fleet	Gender	Partition	AgeErr	Nsamp	Female Data	Male Data	Female N	Male N
1989	1	1	3	0	2	999	<Mean size values>	<Mean size values>	<N fish>	<N fish>

- Nsamp value is ignored if positive, but a negative value cause the entire observation to be ignored
- Negatively valued mean size entries will be ignored in fitting
- Nfish value of 0 will cause mean size value to be ignored in fitting
- Negative value for year causes observation to not be included in the working matrix
- Each gender's data vector and N fish vector has length equal to the number of age' bins.
- Where age data are being entered as conditional age'-at-length and growth parameters are being estimated, it may be useful to include a mean length-at-age vector with nil emphasis to provide another view on the model's estimates.

8.3.9 Environmental data

SS accepts input of time series of environmental data. Parameters can be made to be time-varying by making them a function of one of these environmental time series.

Parameter values can be a function of an environmental data series

2	N environmental variables
10	N environmental observations

Example of 2 environmental observations

Year	Variable	Value
1990	1	0.1
1991	1	0.15

- Any years for which environmental data are not read are assigned a value of 0.0.
- It is permissible to include a year that is one year before the start year in order to assign environmental conditions for the initial equilibrium year. But this works only for recruitment parameters, not biology or selectivity parameters.
- Environmental data can be read for up to 100 years after the end year of the model. Then, if the recruitment-environment link has been activated, the future recruitments will be influenced by any future environmental data. This could be used to create a future "regime shift" by setting historical values of the relevant environmental variable equal to zero and future values equal to 1, in which case the magnitude of the regime shift would be dictated by the value of the environmental linkage parameter. Note that only future recruitment and growth can be modified by the environmental inputs; there are no options to allow environmentally-linked selectivity in the forecast years.

8.3.10 Generalized size composition data

A new feature with SS_v3 is a generalized approach to size composition information. It was designed initially to provide a means to include weight frequency data, but was implemented to provide a generalized capability. The user can define as many size frequency methods as necessary.

- Each method has a specified number of bins.
- Each method has "units" so the frequencies can be in biomass units or numbers units.
- Each method has "scale" so the bins can be in terms of weight or length (including ability to convert bin definitions in pounds or inches to kg or cm).
- The composition data is input as females then males, just like all other composition data in SS. So, in a two-gender model, the new composition data can be combined gender, single gender, or both gender.
- If a retention function has been defined, then the new composition data can be from the combined discard+retained, discard only or retained only.

2	# N WtFreq methods
25 4	#nbins per method
2 1	#units per each method (1=biomass; 2=numbers)
3 2	#scale per each method ((1=kg; 2=lbs; 3=cm; 4=inches)

0.00001 -1	#mincomp to add to each obs (entry for each method)
40 5	#N observations per wtFreq method

Then enter the lower edge of the bins for each method. The two row vectors shown below contain the bin definitions for methods 1 and 2 respectively:

26	28	30	32	34	36	38	40	42	…	60	62	64	68	72	76	80	90
1	2.5	4	9	…	…	…	…	…	…	…	…	…	…	....	…	…	1

- There is no tail compression option for generalized size frequency data;
- Super-period capability is enabled for generalized size comps beginning with V3.20.
- There are two options for treating fish that in population size bins that are smaller than the smallest size frequency bin.
  - Option 1: By default, these fish are excluded (unlike length composition data where the small fish are automatically accumulated up into the first bin.
  - Option 2: If the first size bin is given a negative value, then: accumulation is turned on and the negative of the entered value is used as the lower edge of the first size bin;
- By choosing units=2 and scale=3, the size comp method can be nearly identical to the length comp method if the bins are set identically;
- Bin boundaries can be real numbers so obviously do not have to align with population length bin boundaries, SS interpolates as necessary;
- Size bins cannot be defined to be narrower than the population binwidth; an untrapped error will occur;
- Because the transition matrix can depend upon weight-at-length, it is calculated internally for each gender and for each season because weight-at-length can differ between genders and can vary seasonally.

An example observation is below. Note that its format is identical to the length composition data, including gender and partition options, except for the addition of the first column to indicate the size frequency method.

#Method	Yr	Seas	Fleet	Gender	Partition	samplesize	<composition females then males>
1	1975	1	1	3	0	43	<data>

8.3.11 Tag-recapture data

The ability to analyze tag-recapture data has been introduced with SS_v3. Each released tag group is characterized by an area, time, gender and age at release. Each recapture event is characterized by a time and fleet. Because SS fleet's each operate in only one area, it is not necessary to record the area of recapture. Inside the model, the tag cohort is apportioned across all growth patterns in that area at that time (with options to apportion to only one gender or to both). The tag cohort x growth pattern then behaves according to the movement and mortality of that growth pattern. The number of tagged fish is modeled as a negligible fraction of the total population. This means that a tagging event does not move fish from an untagged group to a tagged group. Instead it acts as if the tags are seeded into the population with no impact at all on the total population abundance or mortality. The choice to require assignment of a predominant age at release for each tag group is a pragmatic coding and model efficiency choice. By assigning a tag group to a single age, rather than distributing it across all possible ages according to the size composition of the release group, it can be tracked as a single diagonal cohort through the age x time matrix with minimal overhead to the rest of the model. Tags are considered to be released at the beginning of a season (period).

1		#Do_Tags If this value is 0, then omit all entries below
COND =1 All subsequent tag-recapture entries must be omitted if Do_Tags = 0.		
	3	#N tag groups
	12	#N recap events
	2	#Mixing latency period: N periods to delay before comparing observed to expected recoveries (0 = release period)
	10	#Max periods (seasons) to track recoveries, after which tags enter accumulator
	#Release Data	

	#TG	area	yr	season	<tfill>	gender	Age	Nrelease
	1	1	1980	1	999	0	24	2000
	2	1	1995	1	999	1	24	1000
	3	1	1985	1	999	2	24	10
	#Recapture Data							

	#TG	year	season	fleet	Number
	1	1982	1	1	7
	1	1982	1	2	5
	1	1985	1	2	0
	2	1997	1	1	6
	2	1997	2	1	4
	3	1986	1	1	7
	3	1986	2	1	5

Note: the release data must be entered in TG order.

Note: <tfill> values are placeholders and are replaced by program generated values for model time.

8.3.12 Stock composition data

It is sometimes possible to observe the fraction of a sample that is composed of fish from different stocks. These data could come from genetics, otolith microchemistry, tags or other means. The growth pattern feature in SS allows definition of cohorts of fish that have different biological characteristics and which are independently tracked as they move among areas. SS now incorporates the capability to calculate the expected proportion of a sample of fish that come from different growth patterns. In the inaugural application of this feature, there was a 3 area model with one stock spawning and recruiting in area 1, the other stock in area 3, then seasonally the stocks would move into area 2 where stock composition observations were collected, then they moved back to their natal area later in the year.

Stock composition data can be entered in SS as follows:

1		#Do morphcomp (if zero, then do not enter any further input below)
COND = 1		
	3	#N observations
	2	#N stocks
	0.00001	#Mincomp

	#Year	Seas	Fleet	partition	Nsamp	Data Vector		
	1980	1	1	0	36	.4	0.6	…
	1981	1	1	0	40	.44	0.62	….
	1982	1	1	0	50	.49	0.50	…

- The N stocks entered with these data must match the N growth patterns in the control file.
- The expected value is combined across genders.
- The “partition” flag is included here in the data, but cannot be used because the expected value is calculated before the catch is partitioned into discard and retained components.
- Note that there is a specific value of mincomp to add to all values of observed and expected.

End of Data

999	#end of data file marker

8.3.13 Excluding data

Data that are <styr or > retroyr are not moved into the internal working arrays at all. So if you have any alternative observations that are used in some model runs and not in others, you can simply give them a negative year value rather than having to comment them out and revise the observation read counter. The first output to data.ss_new has the unaltered and complete input data. Subsequent reports to data.ss_new produce expected values or bootstraps only for the data that are being used. Note that the Nobs values are adjusted accordingly.

Data that are to be included in the calculations of expected values, but excluded from the calculation of –logL, are flagged by use of a negative value for fleet ID.

8.3.14 Data Super Periods

The “Super-Period” capability allows the user to introduce data that represent a blend across a set of time steps and to cause the model to create a expected value for this observation that uses the specified set of time steps. The option is available for all types of data and a similar syntax is used. The syntax is revised for V3.23. Previously, super-periods were started with a -9999 flag in a se or Nsamp field and then stopped with a -9998 flag in that field. This was cumbersome and did not allow for super-periods with only 2 time periods. With 3.23, super-periods are started with a negative value for season, and then stopped with a negative value for season, placeholder observations within the superperiod are designated with a negative fleet field. The se or Nsamp field is now used for weighting of the expected values. An error message will be generated if the old syntax is used. Similarly, negative fleet is the sole allowable flag for omitting observations from the logL calculation. An error message is generated if the superperiod does not contain exactly one observation with a positive fleet field.

All super-period observations must be contiguous in the data file. All but one of the observations in the sequence will have a negative value for fleet ID so the data associated with these dummy observations will be ignored. The observed values must be combined outside of the model and then inserted into the data file for the one observation with a positive fleet ID.

An expected value for the observation will be computed for each selected time period within in the super-period. Beginning with V3.23b, the expected values are weighted according to the values entered in the se (or Nsamp) field for all observations expect the single observation holding the combined data. The expected value for that year gets a relative weight of 1.0. So in the example below, the relative weights are: 1982, 1.0 (fixed); 1983, 0.85; 1985, 0.4; 1986, 0.4. These weights are summed and rescaled to sum to 1.0, and are output in the echoinput.sso file.

Not all time steps within the extent of a super-period need be included. For example, in a 3 season model a super-period could be set up to combine information from season 2 across 3 years, e.g. skip over the season 1 and season 2 for the purposes of calculating the expected value for the super-period. The key is to create a dummy observation (negative fleet value) for all time steps, except 1, that will be included in the super-period and to include one real observation (positive fleet value; which contains the real comboined data from all the specified time steps). Example:

Year	Season	Fleet	Obs	Se	Comment
1982	**-2**	3	34.2	.3	Start super-period. This observation has positive fleet value, so is expected to contain combined data from all identified periods of the superperiod. The *se* entered here is use as the *se* of the combined observation. The expected value for the survey in 1982 will have a relative weight of 1.0 (default) in calculating the combined expected value.
1983	2	**-3**	55	.3	In super-period; entered obs is ignored. The expected value for the survey in 1983 will have a relative weight equal to the value in the *se* field (0.85) in calculating the combined expected value.
1985	2	**-3**	88	.4	Note that 1984 is not included in the superperiod. Relative weight for 1985 is 0.4
1986	**-2**	**-3**	88	.4	End super-period

A time step that is within the time extent of the super-period can still have its own separate observation. In the above example, the survey observation in 1984 could be entered as a separate observation, but it must not be entered inside of the contiguous block of superperiod observations. For composition data (which allow for replicate observations), a particular time steps observations could be entered as a member of a superperiod and as a separate observation.

The super-period concept can also be used to combine seasons within a year with multiple seasons. This usage could be preferred if fish are growing rapidly within the year so their effective age selectivity is changing within year as they grow; fish are growing within the year so fishery data collected year round have a broader size-at-age modes than a mid-year model approximation can produce; and it could be useful in situations with very high fishing mortality.

9. Control File

9.1 Overview of Control File

1. Number of growth patterns and sub-morphs
2. Design matrix for assignment of recruitment to area/season/growth pattern
3. Design matrix for movement between areas
4. Definition of time blocks that can be used for time-varying parameters

5. Specifications for mortality, growth and fecundity
6. Natural mortality and growth parameters for each gender x growth pattern
7. Maturity, fecundity and weight-length for each gender
8. Recruitment distribution parameters for each area, season, growth pattern
9. Cohort growth deviation
10. Environmental link parameters for any MG parameters that use a link
11. Time-varying setup for any MG parms that use blocks
12. Seasonal effects on biology parameters (new in v3)
13. Phase for any MG parms that use annual deviations

14. Spawner-Recruitment parameters
15. Recruitment deviations (much revised in v3)

16. Method for calculating fishing mortality (F)
17. Initial equilibrium F for each fleet

18. Catchability (Q) setup for each fleet and survey (expanded options for random devs in v3)
19. Catchability parameters

20. Length selectivity, retention, discard mortality setup for each fleet and survey
21. Age selectivity setup for each fleet and survey
22. Parameters for length selectivity, retention, discard mortality for each fleet and survey
23. Parameters for age selectivity for each fleet and survey
24. Environmental link parameters for any selectivity/retention parameters that use a link
25. Time-varying setup for any selectivity/retention parameters that use blocks
26. Phase for any selectivity/retention parameters that use random annual deviations

27. Tag-recapture parameters

28. Variance adjustments
29. Lambdas for likelihood components

9.2 Parameter Line Elements

A primary role of the SS control file is to define the parameters to be used by the model. The general syntax of a parameter line is described here. Parameter lines will be used in three sections of the control file: (1) natural mortality and growth; (2) spawner-recruitment, initial F and catchability; and (3) selectivity. The first seven elements of a parameter line are used in every section and will be referred to as a short parameter line. The remaining elements are used just in sections (1) and (3). Each parameter line contains:

Column	Element	Description
1	LO	Minimum value for the parameter
2	HI	Maximum value for the parameter
3	INIT	Initial value for the parameter. If the SS3.PAR file is read, it overwrites these INIT values.
4	Prior Value	Expected value for the parameter. This value is ignored if the Prior_type is –1 or 1
5	Prior type	-1 = none, 0=normal, 1=symmetric beta, 2=full beta, 3=lognormal, 4=lognormal with bias adjustment, 5=gamma
6	Prior stddev	Standard deviation for the PRIOR, used to calculate likelihood of the current parameter value. This value is ignored if Prior_type is –1
7	PHASE	Phase in which parameter begins to be estimated. A negative value causes the parameter to retain its INIT value (or value read from PAR file).
Short parameter lines have only the above 7 elements. The full parameter line syntax for the Mortality-Growth and Selectivity sections provides additional controls to give the parameter time-varying properties. These are listed briefly below and described in more detail in the section Time Varying Parameter Options found at the end of the control file syntax section.		
8	ENV	Create a linkage to an input environmental time series
9	USE_Dev	Invokes use of the dev vector
10	DEV min yr	Beginning year for the dev vector
11	DEV max yr	Ending year for the dev vector
12	DEV std.dev.	Standard deviation for elements in the dev vector.
13	USE-BLOCK	Set up blocks or parameter trends
14	BLOCK-TYPE	Functional form for the block offset

9.3Control File Syntax

The control file is described here using a rather complex set-up with 2 seasons, 2 areas, 2 growth morphs, 2 genders, and 3 sub-morphs in order to demonstrate the order and interdependence of various factors.

Terminology:

- Where the term "COND" appears in the value column of this documentation (it does not actually appear in the control file), it indicates that the following section is omitted except under certain conditions, or that the factors included in the following section depend upon certain condition s.
- In most cases, the description in the Definition column is the same as the label output to the control.ss_new file

VALUE		DESCRIPTION	Comments and Options
#C comment		Comments beginning with #C at top of file will be retained and included in output files	
2		N growth patterns (GP)	
3		N sub-morphs within growth pattern	Permissable values are 1, 3, 5 only. Typical value is 1. Values of 3 or 5 allow exploration of size-dependent survivorship.
COND > 1		Following 2 lines are conditional on N sub-morphs>1	
	0.7	Morph between/within stdev ratio	Ratio of the amount of growth variability between sub-morphs to within sub-morphs
	0.2 0.6 0.2	Distribution among sub-morphs	Enter custom vector or enter –1 for first value of vector to get a normal approx: { 0.15,0.70,0.15} for 3 sub-morphs { 0.031, 0.237, 0.464, 0.237, 0.031} for 5 sub-morphs
COND		Only read recruitment assignment section below if N_GP*N_seas*N_area>1	
	4	Number of recruitment assignments	This specifies which GP, area, seasons have any fraction of the total recruitment assigned. Value here does not affect the required number of recruitment parameter reads below (which must equal N_GP + N_area +N_seas)
	0	Recruitment interaction requested?	If value is 1, then also read N_GP*N_area*N_seas parameters
	1 1 1	Recruitment assignment to GP 1, seas 1, area 1	
	2 1 2	Recruitment assignment to GP 2, seas 1, area 2	
	2 1 2	Recruitment assignment to GP 1, seas 1, area 1	
	2 2 2	Recruitment assignment to GP 2, seas 2, area 2	
COND		Movement: Only read following movement section if N_area>1	
	4	N movement definitions	

	0.6	First age that moves	Real, not integer, age at begin of season. This control primarily used to keep new recruits from moving until after their first year.
	1 1 1 2 4 10	The four requested movement definitions appear here. Each definition specifies: seas, GP, source area, destination area, min age, max age	The rate of movement will be controlled by the movement parameters later. Here the minage and maxage controls specify the range over which the movement parameters are active.
	1 1 2 1 4 10		
	1 2 1 2 4 10		
	1 2 2 1 4 10		
3		N block patterns	These patterns can be referred to in the parameter sections to create a separate parameter value for each block
COND	Following inputs are omitted if N Block patterns equals 0		
	3 2 1	Blocks per pattern	
	1975 19851986 1990 1995 2001	beginning and ending years for blocks in design 1; years not assigned to a block period retain the baseline value for a parameter that uses this pattern	
	1987 1990 1995 2001	beginning and ending years for blocks in design 2	
	1999 2002	beginning and ending years for blocks in design 3	

9.3.1 Biology

This section controls the biology parameters. These include: natural mortality, growth, maturity, fecundity, distribution of recruitment, and movement. Collectively, these are referred to as the MG parameters. The top of the biology section includes several factors that control the number of parameters to be subsequently read and the method by which SS will use these parameters.

0.5		Fraction female	A constant that applies to all growth patterns
1		natM option	0: For 1Parameter; 1: for N_breakpoints; 2: for Lorenzen; 3: read age specific M and do not do seasonal interpolation; 4: read age specific and do seasonal interpolation. Options 1 and 2 also do seasonal interpolation, if appropriate. Each option has different additional inputs below.
COND=0		No additional natM controls	
COND=1			
	4	Number of breakpoints	A value of 2 would correspond to the M pattern as defined in SS V2. Then read a vector of ages for these breakpoints (e.g. corresponding to natM_amin and natM_amax in SS V2). Later, per gender x GP, read N parameters for the natural mortality at each breakpoint.
	2.0 4.5 9.0 15.0	Vector of age breakpoints	
COND = 2			

	4	Read one additional value that is the reference age (integer) for the Lorenzen function	Later read one parameter for each gender x GP that will be the M at the reference age. Other ages will have an M scaled to its body size-at-age. However, if sub-morphs are used, all will have the same M as their growth pattern. Lorenzen M calculation will be updated if the starting year growth parameters are active, but if growth parameters vary during the time series, the M is not further updated. So be careful in using Lorenzen when there is time-varying growth.
COND= 3 or 4		Do not read any natural mortality parameters. With option 3, these M values are held fixed for the integer age (no seasonality or birth season considerations). With option 4, there is seasonal interpolation based on real age, just as in options 1 and 2.	
	0.2 0.25 etc 0.2 0.23 etc	Age-specific M values	Row 1 is female, GP1, row 2 is female GP2, row 3 is male GP1, etc.
1		Growth model	1: von Bertalanffy (2 parameters); 2: Schnute's generalized growth curve (aka Richards curve) with 3 parameters 3: von B, with age-specific K deviations for specified range of ages
1.66		Growth_Amin (A1)	Reference age for first size-at-age parameter .See growth notes below
25		Growth_Amax	Reference age for second size-at-age parameter
COND = 3 Growth option age-specific K			
	5	Min age for age-specific K	
	7	Max age for age-specific K	
0		SD_add_to_LAA	Enter 0.1 to mimic SS2 V1.xx. See growth notes. Recommend using a value of 0.0
1		CV_Pattern	0: CV=f(LAA), so the 2 parameters are in terms of CV of the distribution of length-at-age and the interpolation between these 2 parameters is a function of mean length-at-age; 1: CV=f(A), so interpolation is a function of age; 2: SD=f(LAA), so parameters define the standard deviation of length-at-age and interpolation is a function of mean length-at-age 3: SD=f(A) 4: Lognormal distribution of size-at-age. Input parameters will specify the standard deviation of loge size at age. E.g. entered values will typically be between 0.05 and 0.15. A bias adjustment is applied so the lognormal distribution of size-at-age will have the same mean size as when a normal distribution is used.

2		Maturity option	1: for length logistic; 2: for age logistic; 3: read age-maturity for each female GP;4: read age-fecundity for each female GP. This is a new option in SS_v3. 5: read empirical age-fecundity and body weight-at-age from separate file, wtatage.ss. Allows for reading time series of input. See section Empirical Wt-at-Age for details. NOTE: need to read 2 parameters even if option 3, 4, or 5 is selected 6: read an empirical length-maturity vector by population length bins (available in v3.24q)
COND = 3 or 4 Maturity Option			
	0.0 0.05 0.1 ….	vector of age-specific maturity or fecundity	One row of length Nages+1 for each female GP
COND =6 Maturity Option			
	0.0 0.05 0.1 ….	vector of length specific maturity	One row of length of the population length bins for each female GP
1		First Mature Age	Overridden if maturity option = 3, 4 or 5 but still must exist here
1		Fecundity option	1: to interpret the 2 egg parameters as linear eggs/kg on body weight (current SS default), so fecundity = wt * (a+b*wt), so value of a=1, b=0 causes eggs to be equiv to spawning biomass 2: to set fecundity=a*L^b 3: to set fecundity=a*W^b, so values of a=1, b=1 causes fecundity to be equiv to spawning biomass 4: fecundity =a+b*L; 5: E=a+B*W; This option is irrelevant if Maturity_Option =4 or 5
0		Hermaphroditism option	0 = no; 1 = invoke female to male transition
COND = 1		Read 2 lines below if herma. is selected; Also read 3 parameters after reading the male weight-length parameters	
	-1	Hermaphroditism season	-1 to do transition at end of each season (after mortality and before movement); <positive integer> to select just one season
	1	Include males in spawning biomass	0 = no males in spawnbio; 1=simple add of males to females; xx – more options to come later

2	Offset method	1: direct assignment; 2: for each GP x gender, parameter defines offset from gender 1; • Offsets are in exponential terms, so for example, old_male M = old_female M * exp(old_male parameter) 3: for each GP x gender, parameter defines offset from GP 1 gender 1 • For females, given that "natM option" is breakpoint and there's two breakpoints, parameter defines offset from early age (e.g., old_female_M = young_female_M * exp(old_female_M_parameter). • For males, given that "natM option" is breakpoint and there's two breakpoints, parameter is defined as offset from females AND from early age (e.g., old_male_M = young_female_M * exp(young_male_M_parameter) * exp(old_male_M_parameter))
1	Time-varying adjustment constraint	1: parameter adjustments for env, block and dev are not constrained by bounds 2: parameter adjustments use a logistic transformation to assure that adjusted parameter value stays within bounds of base parameter

9.3.2 Read Mortality-Growth Parameters

Next, SS reads the MG growth parameters.

Parameters	Description
N natM parameters, 3or 4 growth parameters, 2 CV parameters	natural mortality & growth for female, GP1; where the "N" number of natM parameters depends on the option selected. See detailed description of these parameters in section below
"	natural mortality & growth for female, GP2; Note that the order of these blocks of parameters by GP and gender was incorrectly specified in the SS2 user manual.
"	natural mortality & growth for male, GP1
"	natural mortality & growth for male, GP2
2 wt-len, 2 maturity, 2 fecundity	Female biology
2 wt-len	Male biology (if 2 genders exist)
3 hermaphroditism	Only if hermaphroditism is selected
N GP	Recruitment apportionment main effect
N areas	Recruitment apportionment
N seasons	Recruitment apportionment
N patterns x N areas x N seasons	Only if Recr_Dist_Interaction =1 (on). Note that the order of recr_dist parameters has area then seas for main effects, and seas then area for interactions
1	Cohort growth deviation

2 x N selected movement pairs	Movement parameters
7	Ageing error (only if requested in data file age error section)

The biology parameters are:

Natmort_young	Natural mortality for ages <= NMyoung (units are per year)
Natmort_old	Natural mortality for ages >= NMold. For intermediate ages, do a linear interpolation of NM on age.
Lmin	Body length at Amin (units in cm)
Lmax	Body length at Amax (units in cm)
VBK	Von Bertalanffy growth coefficient (units are per year)
COND if growth type = 2	
Richards coefficient	Only include this parameter if Richards growth function is used. If included, a parameter value of 1.0 will have a null effect and produce a growth curve identical to Bertalanffy.
COND if growth type = 3 (age-specific K)	
K deviation for first age in range	
K deviation for next age in range	
…	
K deviation for last age in range	
CV-young	Variability for size at age at age<=AFIX (units are fraction). Note that CV cannot vary over time, so do not set up an env-link or a dev vector. Also, units are either as CV or as stddev, depending on assigned value of CV_pattern.
CV-old	Variability for size at age at age>=AFIX2. For intermediate ages, do a linear interpolation of CV on mean size-at-age. Note that the units for CV will depend on the CV_pattern and the value of Mgparm_as_offset.
Female_scale	coefficient to convert L in cm to Wt in kg
Female_exp	Exponent in female L-W conversion
Mat_inflect	Maturity logistic inflection (in cm or years). Where Female Maturity-at-length (or age) is a logistic function: mat=1/(1 + exp(slope*(<size or age>-inflection)))
Mat_slope	Logistic slope (must have a negative value)
Alpha	Two fecundity parameters; usage depends on the selected Fecundity option. Must be included here even if vector is read in the control section above
Beta	
Male_scale	Male body weight at length parameters. Only include these in a 2 gender model.
Male_exp	
Hermaphrodite inflection age	3 parameters that define a normal distribution for the transition rate of females to males. Only include if hermaphroditism is selected.
Hermaphrodite standard deviation (in age)	
Hermaphrodite asymptotic rate	

9.3.3 Natural Mortality Notes

The options for natural mortality have been expanded. In addition, M is now, in most options, calculated according to real age since the beginning of a cohort's birth season, rather than annual, integer age. So, if M varies by age, M will change by season and cohorts born in early seasons of the year will have different M than late born cohorts.

9.3.4 Growth Notes

When fish recruit at the real age of 0.0 at the beginning of their birth season, they have body size equal to the lower edge of the first population size bin. Previously, they recruited at a size equal to the lower edge of the smallest data size bin. The fish then grow linearly until they reach a real age equal to the input value "growth_age_for_L1" and have a size equal to the parameter value for L1. As they age further, they grow according the Bertalanffy growth equation. The growth curve is calibrated to go through the size L2 when they reach the age "Growth_age_for_L2".

If "Growth_age_for_L2" is set equal to 999, then the size L2 is used as L_{inf}.

If MGparm_def option ==1 (direct estimate, not offsets), then setting a male growth or natural mortality parameter value to 0.0 and not estimating it will cause SS to use the corresponding female parameter value for the males. This check is done on a parameter, by parameter basis and is probably most useful for setting male L1 equal to female L1, then letting males and females have separate K and Linf parameters.

Schnute growth function

The Schnute implementation of a 3-parameter growth function is invoked by entering 2 in the *grow_type* field. Then a fourth parameter is read after reading the Bertalanffy K parameter. When this fourth parameter has a value of 1.0, it is equivalent to the standard von B growth curve. When this function was first introduced in SS, it required that A0 be set to 0.0.

Age-Specific K (beta)

A new growth option, #3, has been introduced in V3.23. This option creates age-specific K deviations for each age of a user-specified age range, with independent additive deviations for each age in the range and for each growth pattern / gender. Each of these deviations is entered as a full parameter line, so inherits all time-varying capabilities of full parameters. The lower end of this age range cannot extend younger than the specified age for which the first growth parameter applies. This is a beta model feature, so examine output closely to assure you are getting the size-at-age pattern you expect. Beware of using this option in a model with seasons within year because the K deviations are indexed solely by integer age according to birthyear. There is no offset for birthseason timing effects, nor is there any seasonal interpolation of the age-varying K.

9.3.4 Growth patterns (morphs) and sub-morphs

The user specifies a number of growth patterns (usually just 1), a number of genders (usually 2), and a number of birth seasons. The collection of Bio_pattern x Gender x BirthSeas constitute the "morphs". The number of sub-morphs per morph can be 1, 3, or 5. The fraction of recruits that are female is specified as an input value (not a parameter), and the fraction of recruits assigned to each sub-morph is custom-input or designated to be a normal approximation. When multiple sub-morphs are designated, an additional input is the ratio of between sub-morph to within sub-morph variability in size-at-age. This is used to partition the total growth variability. Growth parameters are read for each growth pattern x gender combination. For the sub-morphs, their size-at-age is calculated as a factor (determined from the between-within variability calculation) times the size-at-age of the central morph which is determined from the growth parameters for the growth pattern x gender.

9.3.5 Recruitment, Age, and Growth

Recruitment can occur in any season. There still is just one value of spawning biomass calculated annually at the beginning of one specified spawning season and this spawning biomass produces one annual total recruitment value. This annual recruitment is distributed among seasons, areas, and growth types according to other model parameters. These distribution parameters can be time-varying, so the fraction of the recruits that occur in a particular season can change from year to year. For the recruitment apportionment, the parameter values are the ln(apportionment weight), so should have values ranging from about –4 to +4. The product of all apportionment weights is calculated for each pattern x area x season cell that has been designated to receive recruits in the recruitment design matrix. Then the apportionment weights are scaled to sum to 1.0 (within year, not within season) so that the total annual recruitment is distributed among the cells designated to receive recruitment.

In a seasonal model, all cohorts graduate to the age of 1 when they first reach January 1, even if the seasonal structure of the model has them being born in the fall. In general, this means that SS operates under the assumption that all age data have been adjusted so that fish graduate to the next age on Jan 1. This can be problematic if the ageing structures deposit a ring at another time of year. Consequently, you may need to add or subtract a year to some of your age data to make it conform to the SS structure, or you may need to define the SS calendar year to start at the beginning of the season at which ring deposition occurs. Talk with your ageing lab about their criteria for seasonal ring deposition!

Seasonal recruitment is coded to work smoothly with growth. If the recruitment occurring in each season is assigned the same growth pattern, then each seasonal cohort's growth trajectory is simply shifted along the age/time axis. At the end of the year, the early born cohorts will be larger, but all are growing with the same growth parameters so all will converge in size as they approach their common Lmax.

Age 0.0 fish (at beginning of their birth season) are assigned a size equal to the lower edge of the first population size bin and they grow linearly until they reach the age A1. SS generates a warning if the first population length bin is greater than 10 cm as this seems an unreasonably large value for a larval fish. A1 is in terms of real age elapsed since birth. All fish advance to the next integer age on Jan 1, regardless of birth season. For example, consider a 2 season model with some recruitment in each season and with each season's recruits coming from the same GP. At the end of the first year, the early born fish will be larger but both of the seasonal cohorts will advance to an integer age of 1 on Jan 1 of the next year. The full growth curve is still calculated below A1, but the size-at-age used by SS is the linear replacement. Because the linear growth trajectory can never go negative, there is no need for the additive constant to the stddev (necessary for the growth model used in SS2 V1.x), but the option to add a constant has been retained in the model.

9.3.6 Cohort Growth Deviation

This parameter must be given a value of 1.0 and be given a negative phase so it is not estimated. Its importance is in serving as a base for blocks or annual devs, which may be estimated, around this base value of 1.0.

9.3.7 Movement Parameters

There are 2 movement parameters per area pair flagged in the movement design matrix as needing estimable movement parameters. For each, the first parameter is for the movement coefficient for young fish and the second is for old fish (with intermediate ramp calculated using the designated start age and end age. Parameter values are the ln(movement coefficient). For fish that stay in their source area (e.g. move from area 1 to area 1 in season 1), they are given a movement coefficient of ln(1)=0, but this default value is replaced if the stay movement is selected as needed parameters. For each source area, each movement coefficient is exponentiated and then they are scaled to sum to 1.0. At least one needs to not be estimated so that all others are estimated relative to it.

The movement model has been augmented to define movement parameters for each growth pattern. With this capability, it will be possible to have homing of a growth pattern back to its natal area.

An added feature is the reading of migr_firstage immediately after reading the do_migration flag if the do_migration flag is positive. This value is a real number, not an integer, to allow for an in-year start to movement in a multi-season model. The value is the real age at the beginning of a season, even though movement does not occur until the end of the season. For example, in a setup with two 6-month seasons: a value of 0.5 will cause the age 0 fish to not move when they complete their first 6 month season of life, and then to move at the end of their second season because they start movement capability when they reach the age of 0.5 years (6 months).

Note that in developing the firstage capability, a related and currently unresolved logic conflict became apparent. In a multi-season setup, it is possible to allow a growth pattern (GP) to have some fish recruit in different "birthseasons". However, age-specific movement rates are interpolated using integer age, not real age, and movement parameters are common to the entire GP, not to birthseason within GP. Also, movement parameters are explicitly season-specific. Because movement is season-specific, it does not make sense to have movement rates change according to the advancement of real age between seasons. The fix will probably involve making movement parameters be specific to GP x birthseason x actual season.

Future Need: augment the capability further to allow sex-specific movement, and also to allow some sort of mirroring so that genders and growth patterns can share the same movement parameters if desired.

The model will allow movement only between source-destination pairs that have an explicit movement definition. For fish that stay in an area, there are two options:

1. define an explicit movement pattern where the destination area is the same as the source area. This will allow you to control its parameters explicitly;
2. allow the model to create an implicit stay rate definition equivalent to setting the movement strength parameter to 0 for all ages.

For all explicit definitions requested, there must be 2 parameters included with the MG parameter section. As before, the age-specific movement strength is:

1. constant at P1 below minage, constant at P2 above maxage, and linearly interpolated for intermediate ages;
2. exponentiated so that a movement strength parameter value of 0 becomes 1.0;
3. for movement out of an area, the exponentiated value is multiplied by season duration;
4. for each source area, all movement rates are then summed and divided by this sum so that 100% of the fish are accounted for in the movement calculations;
5. it is best if at least one of the destinations for each source area has a predefined movement strength so that other destinations are estimated relative to the fixed value.

9.3.8 Recruitment Allocation and Movement Parameters

In a 2 season, 2 area, 2 growth pattern set-up, the recruitment distribution, cohort growth deviation, and movement parameters could be:

Min	Max	Val	Per standard format…..											Label
-4	4	0	1	1	9	3	0	0	0	0	0.5	0	0	#RecrDist_GP_1
-4	4	0	1	1	9	3	0	0	0	0	0.5	0	0	#RecrDist_GP_2
-4	4	0	1	1	9	3	0	0	0	0	0.5	0	0	#RecrDist_Area_1
-4	4	-4	1	-1	99	3	0	0	0	0	0.5	0	0	#RecrDist_Area_2
-4	4	0	1	1	99	-3	0	0	0	0	0.5	0	0	#RecrDist_Seas_1
-4	4	-4	1	-1	99	-3	0	0	0	0	0.5	0	0	#RecrDist_Seas_2
-1	2	1	1	-1	99	-3	0	0	0	0	0.5	0	0	#CohortGrowDev
-5	5	-4	0	0	99	-5	0	0	0	0	0.5	0	0	#MoveParm_A_seas_1_GP_1from_1to_2
-5	5	-4	0	0	99	-5	0	0	0	0	0.5	0	0	#MoveParm_B_seas_1_GP_1from_1to_2
-5	5	-4	0	0	99	-5	0	0	0	0	0.5	0	0	#MoveParm_A_seas_1_GP_1from_2to_1
-5	5	-4	0	0	99	-5	0	0	0	0	0.5	0	0	#MoveParm_B_seas_1_GP_1from_2to_1
-5	5	-4	0	0	99	-5	0	0	0	0	0.5	0	0	#MoveParm_A_seas_1_GP_2from_1to_2
-5	5	-4	0	0	99	-5	0	0	0	0	0.5	0	0	#MoveParm_B_seas_1_GP_2from_1to_2
-5	5	-4	0	0	99	-5	0	0	0	0	0.5	0	0	#MoveParm_A_seas_1_GP_2from_2to_1
-5	5	-4	0	0	99	-5	0	0	0	0	0.5	0	0	#MoveParm_B_seas_1_GP_2from_2to_1

- For the recruitment parameters, there must be a line for each season, area and GP. But only those seasons, areas, and GPs designated to receive recruits in the recruitment design matrix will have the parameter used in the recruitment distribution calculation.
- For both recruitment allocations and movement rates, SS processes the parameter values according to the following equation:

$$rate_i = e^{p_i} / \sum_{j=1}^{N} e^{p_j}$$

- Set the value of one of these parameters to 0.0 and not estimate it so that other areas will be estimated relative to that base area
- Be sure that estimated parameters are given a min-max of something like -5 and 5 so they have a good range relative to the base area
- In order to get a different distribution of recruitments in different years, you will need to make at least one of the recruitment distribution parameters time-varying.

9.3.9 Ageing error parameters

These 7 parameters are only included in the control file if one of the ageing error definitions in the data file has requested this feature (by putting a negative value for the ageing error of the age zero fish of one ageing error definition. Although these are input with full parameter lines (with inherent time-varying capability), the time-varying updating has not been implemented.

Until a more complete description and examples are developed, here's the code for creation of the vectors of mean age' and stddev of age':

```
for (a=1; a<=nages;a++)
{
   if(r_ages(a)<age_err_parm(1))  // no ageing bias
   {
     age_err(Use_AgeKeyZero,1,a)=r_ages(a)+0.5;
     age_err(Use_AgeKeyZero,2,a)=r_ages(a)/age_err_parm(1)*
                                 age_err_parm(4)+1.0e-5;
   }
   else
   {
      temp=0.0;
         if(r_ages(a)>age_err_parm(1))
            temp=pow((r_ages(a)-age_err_parm(1))/(r_ages(nages)-
            age_err_parm(1)),(age_err_parm(4)));
      age_err(Use_AgeKeyZero,1,a)=(r_ages(a)+0.5)+
            (age_err_parm(2)+temp*(age_err_parm(3)-age_err_parm(2)));
      temp=0.0;
         if(r_ages(a)>age_err_parm(1))
            temp=pow((r_ages(a)-age_err_parm(1))/(r_ages(nages)-
            age_err_parm(1)),(age_err_parm(7)));

      age_err(Use_AgeKeyZero,2,a)=age_err_parm(5)+temp*(age_err_parm(6)
      -age_err_parm(5));
    }
}
```

The 7 parameters are:

1. age at which the estimated pattern begins (just linear below this age). This is "start age"
2. bias at start age (as additive offset from unbiased age')
3. bias at maxage (as additive offset from unbiased age')
4. power fxn coefficient for interpolating between those 2 values (value of 0.0 produces linear interpolation in the bias).
5. stddev at start age
6. stdev at maxage
7. power fxn coefficient for interpolating between those 2 values

9.3.10 Time-varying biology parameters

Any of the parameters defined above can be made time-varying through linkage to an environmental data series, through time blocks, or by setting up annual deviations. The options for making biology and selectivity parameters change over time is detailed in the section labeled Time-Varying Parameters. After reading the biology parameters above, which will include possible instructions to create environmental link, blocks, or dev vectors, then read the following section. Note that all inputs in this section are conditional (COND) on entries in the biology parameter section. So if no biology parameters invoke any time-varying properties, this section is left blank (or completely commented out with #) except for the line with the input of seasonal factors.

9.3.11 Time-varying growth caution

When time-varying growth is used, there are some additional considerations to be aware of:

- Growth deviations propagate into the forecast. The user can select which growth parameters get used during the forecast by setting the end year of the last block. If the last block ends in the model's endyr, then the grorth parameters in effect during the forecast will revert to the "no-block" baseline level. By setting the end year of the last block to endyr+1, the model will continue the last block's growth parameter levels throughout the forecast.
- The equilibrium benchmark quantities (MSY, F40%, etc.) previously used endyr body size-at-age, which is a disequilibrium vector. There is a capability to specify a range of years over which to average the size-at-age used in the benchmark calculations.
- An addition issue occurred in versions prior to 3.20. Its description is retained here, but it was resolved with the growth code modification for version 3.20.
    - Issue for versions prior to 3.20: When the growth reference ages have A1>0 and A2<999, the effect of time-varying K has a non-intuitive aspect. This occurs because the virtual size at age 0.0 and the actual Linf are calculated annually from the current L1, L2 and K parameters. Because these calculated quantities are outside the age range {A1, A2}, a reduction in K will cause an increase in the calculated size-at-age 0.0 that year. So there is a ripple effect as the block's growth parameters affect the young cohorts in existence at the time of the change. The workaround for this is to set A1=0 and A2=999. However, this may create another incompatibility because the size-at-age 0.0 cannot be allowed to be negative and should not be allowed to be less than the size of the first population length bin. Therefore, previous use of A1=2 might have implied a virtual size at age 0.0 that was negative (which is ok), but setting A1=0 does not allow the size at age=A1 to be negative.

Value	Description
COND	If any MG parameters use environmental linkage, then read next factor
0	0: Do not use custom environmental linkage setup, read just one parameter line 1: Use custom environmental linkage, so read one parameter line for each MG parameter that uses linkage

<short parameter line(s)>	Read 0, 1, or many short parameter lines as necessary
COND	If any MG parameters use blocks, then read next factor
0	0: Do not use custom block setup, read just one parameter line 1: Use custom block setup, so read one parameter line for each MG parameter that uses blocks
<short parameter line(s)>	Read 0, 1, or many short parameter lines as necessary
	Seasonality for selected biology parameters (not a conditional input)
0 0 0 0 0 0 0 0 0 0	Read 10 integers to specify which biology parameters have seasonality: femwtlen1,femwtlen2,mat1,mat2,fec1,fec2,Malewtlen1,malewtlen2,L1,K Reading a positive value selects that factor for seasonality. See notes below.
COND	If any factors have seasonality, then read N seasons parameters that define the seasonal offsets from the base parameter value.
<short parameter line(s)>	Read N seasons short parameter lines for each factor selected for seasonality. The parameter values define an exponential offset from the base parameter value.
COND	If any MG parameters use annual deviations, then read the phase next.
-1	All MG parameters using annual deviations will have the deviations begin estimation in this phase

9.3.11.1 Notes on seasonal biology parameters

SS_v3 begins to introduce seasonal effects on selected biology parameters. Currently, seasonal option is only available for the four wt-len parameters and for the growth K. Seasonality is not needed for the maturity and fecundity parameters because spawning is only defined to occur in one season. Seasonal L1 may be implemented at a later date. The seasonal parameter values adjust the base parameter value for that season.

$$P' = P*\exp(seas\_value)$$

9.3.12 Empirical Weight-at-Age

With version 3.04, SS adds the capability to read empirical body weight at age for the population and each fleet, in lieu of generating these weights internally from the growth parameters, weight-at-length, and size-selectivity. Selection of this option is done by setting Maturity_Option = 5. The values are read from a separate file named, wtatage.ss. This file is only required to exist if this option is selected.

The format of this input file is:

syntax for optional input file: wtatage.ss

10		#N rows						
40		#N ages (equal to MaxAge)						
#Yr	seas	gender	GP	birthseas	Fleet	age 0	age 1	…..
-1971	1	1	1	1	1	0.0128586	0.13718	0.432243
-1971	1	1	1	1	2	…..	…..	…..
-1971	1	1	1	1	0	…..	…..	…..

where:

- Fleet = -2 is age-specific fecundity*maturity, so time-varying fecundity is possible to implement
- Fleet = -1 is population wt-at-age at middle of the season
- Fleet = 0 is population wt-at-age at the beginning of the season
- There must be an entry for each fleet for fecundity*maturity, wt-at-age at the middle of the season, and wt-at-age at the beginning of the season.
- GP and birthseas probably will never be used, but are included for completeness
- A negative value for year will fill the table from that year through the ending year of the forecast, overwriting anything that has already been read for those years
- Judicious use of negative years in the right order will allow user to enter blocks without having to enter a row of info for each year
- N ages here equal to maxage specified with the data file, , and N ages +1 columns are required because of age 0 fish.
- If N ages in this table is greater than Maxage in the model, the extra wt-at-age values are ignored
- If N ages in this table is less than Maxage in the model, the wt-at-age for N ages is filled in for all unread ages out to Maxage
- There is no internal error checking to verify that weight-at-age has been read for every fleet and every year
- Fleets that do not use biomass do not need to have wt-at-age assigned
- The values entered for endyr+1 will be used for the benchmark calculations and for the forecast; this aspect needs a bit more checking

CAVEATS:

- SS will still calculate growth curves from the input parameters and can still calculate size-selectivity and can still examine size composition data

- However, there is no calculation of wt-at-age from the growth input, so no way to compare the input wt-at-age from the wt-at-age derived from the growth parameters
- If wt-at-age is read and size-selectivity is used, a warning is generated
- If wt-at-age is read and discard/retention is invoked, then a BEWARE warning is generated because of untested consequences for the body wt of discarded fish.
- Warning: age 0 fish seem to need to have weight=0 for spawning biomass calculation (code -2).

TESTING:

- A model was setup with age-maturity (option 2) and only age selectivity.
- The output calculation of wt-at-age and fecundity-at-age was taken from report.sso and put into wtatage.ss (as shown above)
- Re-running SS with this input wt-at-age (Maturity_Option 5) produced identical results to the run that had generated the weight-at-age from the growth parameters

9.3.13 Spawner-Recruitment

The spawner-recruitment section starts by specification of the functional relationship. The number of parameters needed by each relationship is stored internally (same approach as is used for the number of parameters for each selectivity relationship).

3	Spawner-recruitment specification. The options are: 1: null 2: Ricker (2 parameters) 3: standard Beverton-Holt (2 parameters) 4: ignore steepness and no bias adjustment. Use this in conjunction with very low emphasis on recruitment deviations to get CAGEAN-like unconstrained recruitment estimates. (2 parameters, but only uses the first one.) 5: Hockey stick (3 parameters) for ln(R0), fraction of virgin SSB at which inflection occurs, and the R level at SSB=0.0. 6: Beverton-Holt with flat-top beyond Bzero (2 parameters) 7: Survivorship function (3 parameters). Suitable for sharks and low fecundity stocks to assure recruits are <= pop production

read the required number of short parameter set-up lines. These parameters are:

log(R0)	log of virgin recruitment level
steepness	steepness of S-R; bound by 0.2 and 1.0 for Beverton-Holt
3rd parameter	Optional depending on which function is used
sigma-r	std.dev. of log recruitment; This parameter has two related roles. It penalizes deviations from the spawner-recruitment curve, and it defines the offset between the arithmetic mean spawner-recruitment curve (as calculated from log(R0) and steepness) and the expected geometric mean (which is the basis from which the deviations are calculated. Thus the value of sigmaR must be selected to approximate the true average recruitment deviation.

env-link	environmental linkage coefficient. The recruitment parameters are short parameters, so cannot have the generic block or environmental link options. Instead, this dedicated env-link is provided. It is used to create a multiplicative adjustment to the target parameter, so P_y' = P *exp(env_link * env_data$_y$). An alternative that provides an additive link is under development.
log(R1)	offset for initial equilibrium recruitment relative to virgin recruitment.
autocorrelation	Autocorrelation in recruitment

Then read additional spawner-recruitment conditions:

Value	Label	Description and Options
0	SR_env_link	This is the index of the environmental variable that will be used as the basis for adjustment of SR expectations. This works for both the forecast period and for the initial equilibrium (by entering a value for the environmental variable one year before the start of the time series.)
3	SR_env_target	This factor determines what aspect of spawner-recruitment is affected by the environmental variable. The options are: 1: annual deviations 2: R0 3: steepness If the application needs to compare the environment to annual recruitment deviations, then the preferred option is to transform the environmental variable into an age 0 pre-recruit survey and enter these as a survey with expected value based on selectivity option #31. Use of SR_env_target=1 is discouraged because it interacts with the level of residual recruitment variability and there is no implementation of a bias correction for the variability in recruitment caused by the environmental variable. If the application is investigating regime shifts, then enter an environmental variable with a time series of zeroes and ones to describe the regime periods, then use SR_env_target of 2 or 3 to adjust the expected level of recruitment according to the regime variable. Note that MSY related quantities will be calculated with the regime in the zero state only. However, the forecast can be responsive to designated regime levels.
1	Do_recr_dev	This selects the way in which recruitment deviations are coded: 0: none (so all recruitments come from S-R curve) 1: devvector (previously the only option). Here the deviations are encoded as a dev_vector, so ADMB enforces a sum-to-zero constraint. 2: simple deviations. Here the deviations do not have an explicit constraint to sum to zero, although they still should end up having close to a zero sum. The difference in model performance between options (1) and (2) has not been fully explored to date.
1971	Main recr devs begin year	If begin year is less than the model start year, then the early deviations are used to modify the initial age composition. However, if set to be more than Nages before start year, it is changed to equal Nages before start year
1999	Main recr devs end year	If recr devs end year is later than retro year, it is reset to equal retro year.
3	Main recr dev phase	
1	Advanced Options	0: Use default values for advanced options; 1: Read values for the 11 advanced options
COND = 1 Beginning of advanced options		

	1950	Early recr dev start year	0: skip; +year: absolute year (must be less than begin year of main recr devs) -integer: set relative to main recr dev start year NOTE: because this is a dev vector, it should be long enough so that recr devs for individual years are not unduly constrained. Default: 0
	6	Early recr dev phase	Users may want to set to a late phase if there is not much early data; Default: -4
	0	Forecast recruitment phase	Forecast recruitment deviations always begin in the first year after the end of the main recruitment deviations. Setting their phase to 0 causes their phase to be set to max lambda phase +1 (so that they become active after rest of parameters have converged.). However, it is possible here to set an earlier phase for their estimation, or to set a negative phase to keep the forecast recruitment devs at a constant level. Default: 0
	1	Forecast recr devs lambda	This lambda is for the logL of the forecast recruitment devs that occur before endyr+1. Use a larger value here if solitary, noisy data at end of time series cause unruly recr dev estimation. Default: 1.0
	1956	Last year with no bias adjustment	These four entries control how the bias adjustment is phased in and then phased back out when the model is searching for the maximum logL. Bias adjustment is automatically turned off when in MCMC mode. For intervening years between the first and second years in this list, the amount of bias adjustment that will be applied is linearly phased in. The first year with full bias adjustment should be a few years into the data-rich period so that SS will apply the full bias-correction only to those recruitment deviations that have enough data to inform the model about the full range of recruitment variability. See the recruitment advisory for more information. Defaults for the four year values: Start year – 1000. Start year – Nages Main recr dev final year End year +1.
	1970	First year with full bias adjustment	
	2001	Last year with full bias adjustment	
	2002	First recent year with no bias adjustment	
	0.85	Max bias adjustment	Value for the maximum bias adjustment during the MLE mode. Use value of 1.0 for compatibility with previous versions of SS. All estimated recrdevs, even those within a ramped era, switch to maxbias=1.0 during MCMC.
	0	Period for recruitment cycles	Use this when SS is configured to model seasons as years and there is a need to impose a periodicity to the expected recruitment level. If value is >0, then read that number of full parameter lines below define the recruitment cycle
	-5	Min recr dev	Min value for recruitment deviation; Default: -5
	5	Max recr dev	Max value for recruitment deviation; Default: +5
	2	N explicit recr devs to read	0: Do not read any recruitment deviations; Integer: read this number of recruitment deviations; Default: 0
END OF ADVANCED OPTIONS			
COND = Enter N full parameter lines below if N recruitment cycles is >0			
	<full (e.g. 14 element) parameter line for each of the N periods of recruitment cycle>		
COND = If N explicit recr devs is >0, then enter N lines below			

	1977 3.0	Year, deviation	First of two recruitment deviations being read. NOTE: SS will rescale the entire vector of recrdevs after reading these deviations, so by reading two positive values, all other recrdevs will be scaled to a small negative value to achieve a sum to zero condition before starting model estimation
	1984 3.0	Year, deviation	

9.3.13.1 Spawner-Recruitment Functions

The number of age-0 fish is related to spawning biomass according to a stock-recruitment relationship. SS has the option of the Beverton-Holt, Ricker, Hockey-Stick, and a survival-based stock recruitment relationship.

Beverton-Holt

The Beverton-Holt Stock Recruitment curve is calculated as:

$$R_y = \frac{4hR_0 SB_y}{SB_0(1-h)+SB_y(5h-1)} e^{-0.5 b_y \sigma_R^2 + \tilde{R}_y} \qquad \tilde{R}_y \sim N\left(0;\sigma_R^2\right)$$

where R_0 is the unfished equilibrium recruitment, SB_0 is the unfished equilibrium spawning biomass (corresponding to R_0), SB_y is the spawning biomass at the start of the spawning season during year y, h is the steepness parameter, b_y is the bias adjustment fraction applied during year y, σ_R is the standard deviation among recruitment deviations in log space, and $\tilde{R}_y$ is the lognormal recruitment deviation for year y. The bias-adjustment factor (Methot and Taylor 2011) ensures unbiased estimation of mean recruitment even during data-poor eras in which the maximum likelihood estimate of the $\tilde{R}_y$ is near 0.0.

Ricker

The Ricker Stock Recruitment curve is calculated as:

$$R_y = \left(\frac{R_0 SB_y}{SB_0}\right) e^{h\left(1-SB_y/SB_0\right)} e^{-0.5 b_y \sigma_R^2 + \tilde{R}_y} \qquad \tilde{R}_y \sim \tilde{N}\left(0;\sigma_R^2\right)$$

Hockey-Stick

The hockey-stick recruitment curve is calculated as:

$$R_y = R_{\min} R_0 + \left(\frac{SB_y}{hSB_0}\right)\left(R_0 - R_{\min}\right)\left(join\right) + R_0\left(1 - join\right)$$

$$join = \left[1 + e^{\left(1000*\left(SB_y - hSB_0\right) / SB_0\right)}\right]^{-1}$$

where $R_{\min}$ is the miminum recruitment level predicted at a spawning size of zero and is set by the user in the control file, and h is defined as the fraction of SB_0 below which recruitment declines linearly.

Survivorship

Survival-based recruitment (Taylor et al. 2012) is constrained so that the recruitment rate cannot exceed fecundity

$$R_y = e^{\left(-z_0+\left(z_0-z_{\min}\right)\left(1-\left(SB_y/SB_0\right)^{\rho}\right)\right)} SB_y e^{-0.5 b_y \sigma_R^2 + \tilde{R}_y} \qquad \tilde{R}_y \sim \tilde{N}\left(0;\sigma_R^2\right)$$

where z_0 (P) is the negative log of the pre-recruit mortality rate at unfished equilibrium, z_{min} is the limit of the pre-recruit mortality as relative spawning biomass approaches 0, parameterized as a function of z_{frac} (P) (which represents the reduction in mortality as a fraction of z_0), and ρ (P) is a parameter controlling the shape of density-dependent relationship between relative spawning biomass and pre-recruit survival. The steepness (h) of the spawner-recruit curve (defined as recruitment relative to R_0 at a spawning depletion level of 0.2) is:

$$h = 0.2e^{z_0 z_{frac}\left(1-0.2^{\beta}\right)}$$

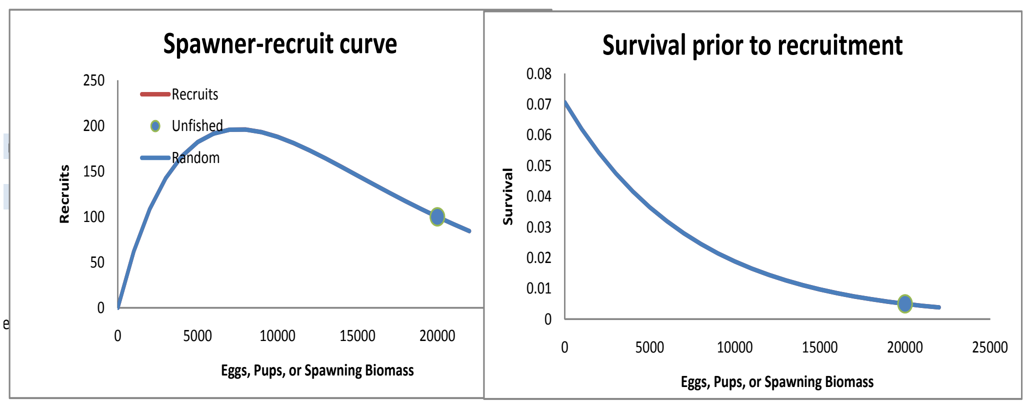
This 3-parameter function was created for use with low fecundity species, but its use of 3-parameters provides a flexibility comparable to the 3-parameter Shepard function. This survival based spawner-recruitment function defines survival from the egg (e.g. hatched pups) to the recruits stage to be a declining function of the initial number of pups produced (Taylor et al. 2012).

- Start with the parameter, ln(R_0), which is the ln(mean number of recruits) that enter the population in unfished conditions.
- These recruits over their lifetime will produce some total number of eggs (pups), termed Pups_0, which can be calculated from natural mortality, which defines the numbers at age in the adult population, and fecundity at age.
- Because the unfished condition is considered to be a stable equilibrium, we can calculate PPR_0 = Pups_0/R_0 and its inverse which is survivorship, which we will define in logarithmic space. So, Z_0 = ln(R_0/Pups_0). Note that there is no explicit time over which this Z acts. Such an explicit time (e.g. the age ar recruitment) may be implemented in the future. For now, this means that the Z is really a Z*delta t.
- So, Z_0 is the survival when the population is at carrying capacity. On the other extreme, the maximum survival is 1.0, so the maximum Z is 0.0.
- The parameter, S_frac, defines the level of Z when the population approaches an abundance of 0.0. This has values bounded by 0.0 and 1.0 and creates a Z_max which is between Z_0 and 0.0.
    - Z_max = Z_0 + S_frac*(0.0-Z_0)
- Then for the current level of pup production (e.g. total population fecundity, aka "spawning biomass"):
    - Z_y=(1 - (Pup_y/Pups_0)^{Beta})*(Z_max-Z_0)+Z_0
    - So R_y = Pup_y * exp(-Z_y)
    - Where beta is the third parameter and which logically has values between about 0.4 for a left-shifted spawner-recruitment curve, and 3.0 for a right-shifted curve.

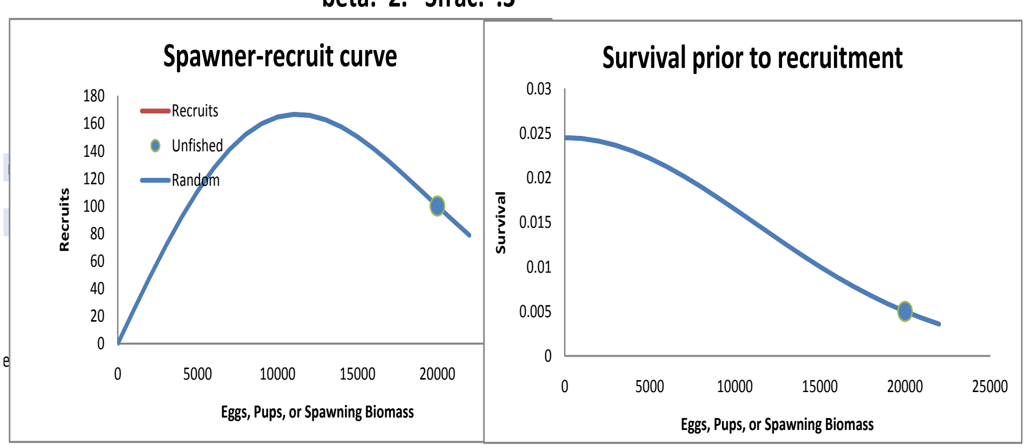
- With the other spawner-recruitment relationships, the mean level of recruits, R_y , serves as the base against which environmental effects and annual lognormal deviations are applied. However, in a survival context, it is possible that a large positive deviation on recruitments could imply survival greater than 1.0, so an alternative approach is needed for this survival approach. Here, the lognormal deviations are applied to Z and the resultant S is constrained to not exceed 1.0.
- In SS, it is also necessary to be able to calculate the equilibrium level of spawning biomass (pup production) and recruitment for a given level of spawning biomass per recruit (pups per recruit), PPR.
    - Pups_equil = Pups_0 * (1 - (LN(1/PPR) - Z_0)/(Z_max - Z_0)) $^{(1/\text{Beta})}$
    - Then, R_equil = Pups_equil * exp(-(1 - (Pups_equil/Pups_0)$^{\text{Beta}}$)*(Z_max-Z_0)+Z_0)

Some example plots for various levels of S_frac and beta are shown below.

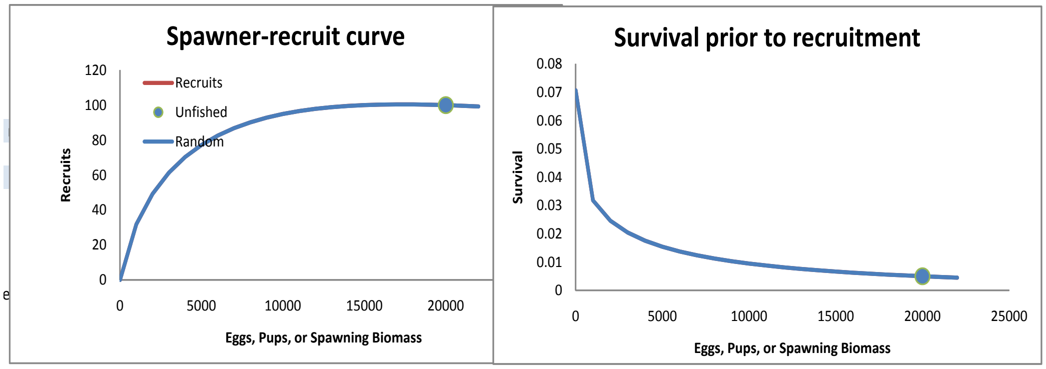
beta: 1. Sfrac: .5



beta: 2. Sfrac: .3



beta: .4 Sfrac: .5




```
case 7:  // survival based, so constrained such that recruits cannot
           exceed  fecundity
{
       // PPR_0=SPB_virgin_adj/Recr_virgin_adj;
       // pups per recruit at virgin
       // Surv_0=1./PPR_0;
       // recruits per pup at virgin
       // Pups_0=SPB_virgin_adj;
       // total population fecundity is the number of pups produced
       // Sfrac=SR_parm(2);
            SRZ_0=log(1.0/(SPB_virgin_adj/Recr_virgin_adj));
            SRZ_max=SRZ_0+SR_parm(2)*(0.0-SRZ_0);
            SRZ_surv=mfexp((1.-
                          pow((SPB_curr_adj/SPB_virgin_adj),SR_parm(3))
                          *(SRZ_max-SRZ_0)+SRZ_0);  //  survival
            NewRecruits=SPB_curr_adj*SRZ_surv;
            exp_rec(y)=NewRecruits;      //  expected  arithmetic  mean
                          recruitment
              if(SR_env_target==1)
                 SRZ_surv*=mfexp(SR_parm(N_SRparm2-
                 2)*env_data(y,SR_env_link));     //  environ  effect  on
                 survival
              if(recdev_cycle>0)
              {
               gg=y- (styr+(int((y-styr)/recdev_cycle))*recdev_cycle)+1;
               SRZ_surv*=mfexp(recdev_cycle_parm(gg));
              }
            pred_rec(y)=SPB_curr_adj*SRZ_surv;
              if(y <=recdev_end)
              {
               if(recdev_doit(y)>0) SRZ_surv*=mfexp(recdev(y));
                     //  recruitment deviation
              }
              else if(Do_Forecast>0)
              {
                 SRZ_surv *= mfexp(Fcast_recruitments(y));
              }
            join=1./(1.+mfexp(100*(SRZ_surv-1.)));
            SRZ_surv=SRZ_surv*join + (1.-join)*1.0;
            NewRecruits=SPB_curr_adj*SRZ_surv;
            use_rec(y) = NewRecruits;
            break;
}
```

9.3.13.2 Recruitment eras

Conceptually, SS treats the early, data-poor period, the main data-rich period, and the recent/forecast time period as three eras along a continuum. The user has control of the break year between eras. Each era has its own vector. The early era is defined as a vector (prior to V3.10 this was a dev_vector) so it can have zeros during the earliest years not informed by data and then a few years with non-zero values without imposing a zero-centering on this collection of deviations. The main era can be a vector of simple deviations, or a dev_vector but it is normally implemented as a dev_vector so that the spawner-recruitment function is its central tendency. The last era does not force a zero-centered dev vector so it can have zeros during the actual forecast and non-zero values in last few years of the time series. The early and last eras are optional, but their use can help prevent SS from balancing a preponderance of negative devs in early years against a preponderance of positive devs in later years. When the 3 eras are used, it would be typically to turn on the main era during an early model phase, turn on the early era during a later phase, then have the last era turn on in the final phase.

9.3.13.3 Recruitment Likelihood

In SS2, recruitment log(L) contained a term, + N_forecast_rec_devs*log(sigmaR). This meant that the total log(L) changed according to how many forecast years were included in the model scenario. Worse, if sigmaR was allowed to be estimated by SS2, then this term would cause all the zero devs during the forecast period to drag the overall estimated value of sigmaR down. This problem is rectified in SS V3. Now, for each year in the total time series (early, mid, late/forecast) the contribution of that year to the logL is equal to: dev^2/(2.0*sigmaR*sigmaR)+offset*log(sigmaR); where offset is the magnitude of the adjustment between the arithmetic and geometric mean of expected recruitment for that year. With this approach, years with a zero or small offset value do not contribute to the second component. With this approach, sigmaR may be estimable when there is good data to establish the time series of recruitment deviations. In the likegfish example, turning on estimate of sigmaR results in an estimated value that is very close to the root mean squared error (rmse) of the estimated recruitment deviations.

9.3.13.4 Recruitment bias adjustment

The recruitment bias adjustment implemented in SS is based upon the work being documented in Methot and Taylor (2011) and following the work of Maunder and Deriso (2003). The concept is based upon the following logic. SigmaR represents the true variability of recruitment in the population. It provides the constraining penalty for the estimates of recruitment deviations and it is not affected by data. Where data that are informative about recruitment deviations are available, the total variability in recruitment, sigmaR, is partitioned into a signal (the variability among the recruitment estimates) and the residual, the variance of each recruitment estimate (see eq. below). Where there are no data, no signal can be estimated and the individual recruitment deviations collapse towards 0.0 and the variance of each recruitment deviation approaches sigmaR. Conversely, where there highly informative data about the recruitment deviations, then the variability among the estimated recruitment deviations will approach sigmaR and the variance of each recruitment deviation will approach zero. Perfect data will estimate the recruitment time series signal perfectly. Of course, we never have perfect data so we should always expect the estimated signal (variability among the recruitment deviations) to be less than the true population recruitment variability.

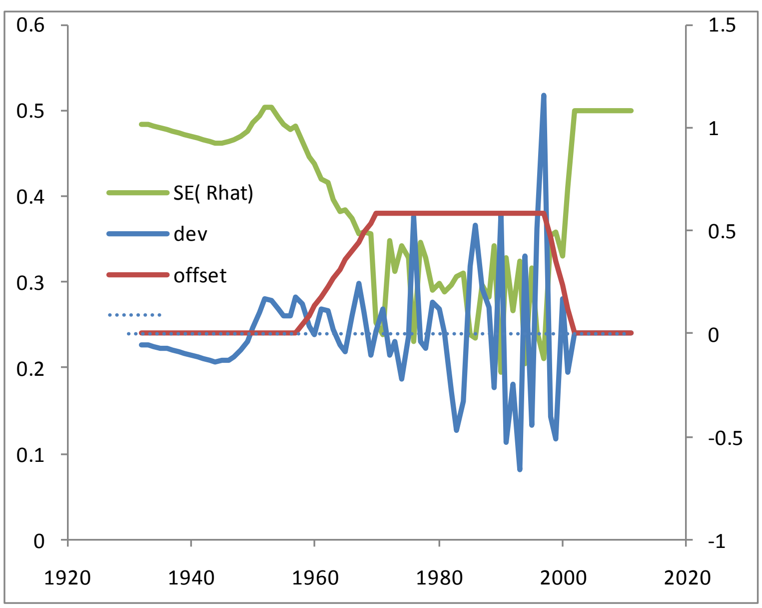
$$\mathrm{SE}(\hat{r}_y)^2 + \mathrm{SD}(\hat{r})^2 = \left(\left(\frac{1}{\sigma_d{}^2} + \frac{1}{\sigma_R{}^2} \right)^{-1/2} \right)^2 + \left(\frac{\sigma_R{}^2}{\left(\sigma_R{}^2 + \sigma_d{}^2\right)^{1/2}} \right)^2 = \sigma_R{}^2.$$

The correct offset (bias adjustment) to apply to the expected value for recruitment is based on the concept that a time series of estimated recruitments should be mean unbiased, not median unbiased, because the biomass of a stock depends upon the cumulative number of recruits, which is dominated by the large recruitments. The degree of offset depends upon the degree of recruitment signal that can be estimated. Where no recruitment signal can be estimated, the median recruitment is the same as the mean recruitment, so no offset is applied. Where lognormal recruitment signal can be estimated, the mean recruitment will be greater than the median recruitment. The value

$$b_y = \frac{\mathrm{E}\left(\mathrm{SD}(\hat{r}_y)\right)^2}{\sigma_R{}^2} = 1 - \frac{\mathrm{SE}(\hat{r}_y)^2}{\sigma_R{}^2}$$

of the offset then depends upon the partitioning of sigmaR into between and within recruitment variability. The most appropriate degree of bias adjustment can be approximated from the relationship among sigmaR, recruitment variability (the signal), and recruitment residual error.

Because the quantity and quality of data varies during a time series, SS allows the user to control the rate at which the offset is ramped in during the early, data-poor years, and



then ramped back to zero for the forecast years.

On output to report.sso, SS calculates the mean bias adjustment during the early and main eras and compares it to the rmse of estimated recruitment devs. A warning is generated if the rmse is small and the bias adjustment is larger than 2.0 times the ratio of $rmse^2$ to $sigmaR^2$.

In MCMC mode, the model still draws recruitment deviations from the lognormal distribution, so the full offset is used such that the expected mean recruitment from this lognormal distribution will stay equal to the mean from the spawner-recruitment curve. When SS reaches the MCMC and MCEVAL phases, all biasadj values are set to 1.0 for all active recruitment deviations because the model is now re-sampling from the full lognormal distribution of each recruitment.

9.3.13.5 Recruitment Autocorrelation

The autocorrelation parameter is implemented. It is not performance tested and it has no effect on the calculation of the offsets described in the section above.

9.3.13.6 Recruitment Cycle

When SS is configured such that seasons are modeled as years, the concept of season within year disappears. However, there may be reason to still want to model a repeating pattern in expected recruitment to track an actual seasonal cycle in recruitment. If the recruitment cycle factor is set to a positive integer, this value is interpreted as the number of time units in the cycle and this number of full parameter lines will be read. The cyclic effect is modeled as an exp(p) factor times R0, so a parameter value of 0.0 has nil effect. In order to maintain the same number of total recruits over the duration of the cycle, a penalty is introduced so that the cumulative effect of the cycle produces the same number of recruits as Ncycles * R0. Because the cyclic factor operates as an exponential, this penalty is different than a penalty that would cause the sum of the cyclic factors to be 0.0. This is done by adding a penalty to the parameter likelihood, where:

$$X = \text{sum}(\exp(p))$$

$$Y = \text{Ncycle}$$

$$\text{Penalty} = 10000*(X-Y)^2$$

9.3.13.7 Initial Age Composition

A non-equilibrium initial age composition is achieved by setting the first year of the recruitment deviations before the model start year. These pre-start year recruitment deviations will be applied to the initial equilibrium age composition to adjust this composition before starting the time series. The model first applies the initial F level to an equilibrium age composition to get a preliminary N-at-age vector, then it applies the recruitment deviations for the specified number of younger ages in this vector. If the number of estimated ages in the initial age composition is less than Nages, then the older ages will retain their equilibrium levels. Because the older ages in the initial age composition will have progressively less information from which to estimate their true deviation, the start of the bias adjustment should be set accordingly.

9.3.14 Fishing Mortality Method

There are now three methods available for calculation of fishing mortality. These are: Pope's approximation, continuous F with each F as a model parameter, and a hybrid method that does a Pope's approximation to provide initial values for iterative adjustment of the continuous F values to closely approximate the observed catch. With the hybrid method, the final values are in terms of continuous F, but do not need to be specified as full parameters. In a 2 fishery, low F case, it is just as fast as the Pope approx. and produces identical result. When F is very high, the problem becomes quite stiff for Pope's and the hybrid method so convergence may slow. It may still be better to use F option 2 (continuous F as full parameters) in these high F cases. F as parameter is also preferred for situations where catch is known imprecisely and you are willing to accept a solution in which the final F values do not reproduce the input catch levels exactly. Option 1 (Pope's approx) still exists, but my recommendation is to switch to option 3.

Value		Label	Description
0.2		F ballpark	This value is compared to the sum of the F's for the specified year. The sum is over all seasons and areas. The lambda for the comparison goes down by a factor of 10 each phase and goes to 0.0 in the final phase.
-1990		F ballpark year	Negative value disables F ballpark
3		F Method	1: Pope's 2: Continuous F as parameters 3: Hybrid
0.9		Maximum F	This maximum is applied within each season and area. A value of 0.9 is recommended for F method 1, and a value of about 4 is recommended for F method 2 and 3.
COND	Depending on F Method		
COND: 1	No additional input for Pope's approx.		
COND: 2			
	0.1	Starting value for each F	Initializing value for each F parameter
	1	Phase for F parameters becoming active	For phases prior to this value, SS will use the hybrid option and the F values so calculated become the starting values for the F parameters when this phase is reached.
	1	Number of detailed F inputs to read below	
COND: 3			
	4	Number of tuning iterations in hybrid method	A value of 2 or 3 is sufficient with a single fleet and low Fs. A value of 5 or so may be needed to match the catch near exactly when there are many fleets and high F.
COND	If N for F detail is >0		
	1, 1980, 1, 0.2, 0.05, 4	fleet, yr, seas, F, se, phase	These values override the catch se values in the data file and the overall starting F value and phase read just above

9.3.15 Initial Fishing Mortality

Read a short parameter setup line for each fishery. The parameters are the fishing mortalities for the initial equilibrium. Do not try to estimate parameters for fisheries with zero initial equilibrium catch. If there is catch, then give a starting value greater than zero and it generally is best to estimate the parameter in phase 1.

It is possible to use the initial F method to achieve an estimate of the initial equilibrium Z in cases where the initial equilibrium catch is unknown. To do this:

- Include a positive value for the initial equilibrium catch;
- Set the lambda for the logL for initial equilibrium catch to a nil value (hence causing SS to ignore the lack of fit to the input catch level;
- Allow the initial F parameter to be estimated. It will be influenced by the early age and size comps which should have some information about the early levels of Z.

9.3.16 Catchability

For each fishery and survey, enter a row with these 4 entries as described below.

1. Do_Power
   a. 0 = skip, so survey is directly proportional to abundance
   b. 1 = establish a parameter for non-linearity in survey-abundance linkage
   c. typical = 0

2. Do_Env_Link (typical = 0)
   a. 0 = skip, no environmental effect on Q
   b. 1 = establish a parameter to create environmental effect on Q, where the integer is the index of the environmental variable to be linked. The relationship is: ln_q(y) = ln_q_base + Q_env_link_parameter*Env_Value(y).

3. Do_extra SD (typical = 0)
   a. 0 = skip
   b. 1 = estimate a parameter that will contain an additive constant to be added to the input stddev of the survey variability. This extra SD approach is highly redundant with the older code that provided for iterative input of variance adjustment factors. The newer code for extra SD estimation is recommended.

4. Q type
   a. <0=mirror the Q from another (lower numbered) survey designated by abs(value)
   b. 0 = set Q as a scaling factor such that the estimate is median unbiased. This is comparable to the old “float” option. This option is not available if a normal error structure is used.
   c. 2 = establish one parameter that will be the ln(Q). Note that Q is in log units even if the error structure is normal.
   d. 3 = establish one parameter that will be the base ln(Q) and a set of additional parameters for each year of the survey that will be deviations in ln(Q). These deviation parameters are full parameters, so each has a prior and variance, so surveys with high uncertainty in their calibration can be given a more diffuse prior to allow a larger deviation. Because each of these Q deviations is coded

as a separate parameter, rather than a member of a dev_vector, the contribution of these deviations to the model's objective function is captured in the parameter prior section. However, because there is no inherent constraint that these deviations have a zero sum, a separate log(L) contribution is calculated from the sum of the devs (=square(1.+square(sum_devs))-1.) and added to the "parm_dev_like" component.

e. 4 = establish one parameter that will be the base ln(Q) and used as the Q for the first survey observation. Subsequent N-1 parameters for remaining survey observations will be deviations in random walk of ln(Q). These deviation parameters are otherwise treated identically to those generated by option (3) above, except that the extra contribution for the mean deviation is not calculated.

f. 5 = This option will calculate the survey Q according to mean unbiased scaling, then assigns this value to the parameter (which must be set up in the control file and be given a negative phase). Advantage is that the calculated Q can now have a prior.

So for a setup with 2 fisheries and 2 surveys, the Q setup matrix could be:

A= do power	B = env-var	C= extra SD	D= devtype (<0=mirror, 0/1=none, 2=cons, 3=rand, 4=randwalk)	
1	0	1	2	#Fish_1
0	0	1	2	#Fish_2
0	0	0	4	#Survey_1
0	0	1	2	#Survey_2

COND		If any fishery or survey uses random devs or random walk, there is an option to either read detailed input to set up the deviations, or to just read a template	
#Value		Label	Description and Options
	1	Read detailed input for random effects	0: read one parameter line and use it as a template to create a time series of parameters for each observation for each fleet/survey that uses random effects. The output to control.ss_new will be in detailed format even if the input is not detailed. Therefore a simple way to create a detailed setup file is to start with a simple template then edit the control.ss_new file to create a detailed input for subsequent model runs; 1: read a parameter line for each observation of each fleet/survey that uses random effects, thus allowing customization. If the Q option for a fleet is 3 (random devs), then read one parameter for each observation. If the Q option is 4, then read (N observations -1) parameters.

For each positive element in columns A – D above, read a short parameter setup line.

The order is: fishery 1 through survey N within power transformation, then within environment link, then within extra stddev, then within Q.

If no elements are selected, then there must be no parameter setup lines.

The list of parameters to be read from the above setup would be:
Fishery 1 power
Fishery 1 extra SD
Fishery 2 extra SD
Survey 2 extra SD
Fishery 1 base Q
Fishery 2 base Q
Survey 1 base Q
Survey 1 Q randwalk for observation 2
Survey 1 Q randwalk for observation 3
Survey 1 Q randwalk for observation 4
Etc.
Survey 3 base Q

9.3.17 Selectivity and Discard

For each fleet and survey, read a definition line for size selectivity and retention. The four values read from each line are:

PATTERN: Valid length selectivity pattern code.

DISCARD: (0/1/2/3 or -index) If value is 1, then program will read 4 retention parameters after reading the specified number of selectivity parameters and all discarded fish are assumed dead. If the value is 2, then the program will read 4 retention parameters and 4 discard mortality parameters. If the value is 3, then no additional parameters are read and all fish are assumed discarded and dead. If the value is a negative number, then it will mirror the retention and discard mortality pattern of the lower numbered fleet.

MALE: (0/1/2/3/4) If value is 1, then program will read 4 additional parameters to define the male selectivity relative to the female selectivity. Anytime the male selectivity is caused to be greater than 1.0; the entire male/female matrix of selectivity values is scaled by the max so that the realized max is 1.0. Hopefully this does not cause gradient problems. If the value is 2, then the main selectivity parameters define male selectivity and female selectivity is estimated as an offset from male selectivity. This alternative is preferable if female selectivity is less than male selectivity. The option 3 is only available if the selectivity pattern is 1, 20, or 24 and it causes the male selectivity parameters to be offset from the female parameters, rather than the male selectivity being an offset from the female selectivity.

SPECIAL: (0/value). This value is used in different ways depending on the context. If the selectivity type is to mirror another selectivity type, then put the index of that source fleet or survey here. It must refer to a lower numbered fleet/survey. If the selectivity type is 6 (linear segment), then put the number of segments here. If the selectivity type is 7, then put a 1 here to keep selectivity constant above the mean average size for old fish of morph 1.

For each fleet and survey, read a definition line for age selectivity. The 4 values to be read are the same as for the size-selectivity. However, the retention value must be set to 0.

#_size_selex_types				
#_Pattern	Discard	Male	Special	
27	0	0	3	#Fishery1
1	0	0	0	#Survey 1
0	0	0	0	#Survey 2
#_age_selex_types				
#_Pattern	Discard	Male	Special	
11	0	0	0	#Fishery1
11	0	0	0	#Survey 1
11	0	0	0	#Survey 2

9.3.18 Selectivity Patterns

The currently defined selectivity patterns, and corresponding required number of parameters, are:

SIZE SELECTIVITY		
Pattern	N Parameters	Description
0	0	Selex=1.0 for all sizes
1	2	Logistic
2	8	Double logistic, with defined peak, uses IF joiners. Discontinued. Use pattern #8 instead.
3	6	Discontinued.
4	0	set size selex=female fecundity. Discontinued. Use pattern #30 instead
5	2	mirror another selex. The two parameters select bin range
15	0	Mirror another selex (same as for age selex)
6	2+special value	Non-parametric
7	8	Double logistic, with defined peak, uses smooth joiners; special=1 causes constant selex above Linf for morph 1. Discontinued. Use pattern #8.
8	8	Double logistic, with defined peak, uses smooth joiners; special=1 causes constant selex above Linf for morph 1.
9	6	Simple double logistic, no defined peak
22	4	Double normal; similar to CASAL
23	6	Same as the double normal pattern #24 except the final selectivity is now directly interpreted as the terminal selectivity value.
24	6	Double normal with defined initial and final selectivity level – Recommended option. Test using SELEX-24.xls. See new notes for SS_v3 below.
25	3	Exponential-logistic
27	3+2*N_nodes	Cubic spline
SPECIAL SELECTIVITY		
Pattern	N Parameters	Description

30	0	Sets expected survey abundance equal to spawning biomass (population fecundity)
31	0	Sets expected survey abundance equal to exp(recruitment deviation). This is useful if environmental data is used as an index of recruitment variability.
32	0	Sets expected survey abundance equal to exp(recruitment deviation) * SpawnBiomass. So this is recruitment without density-dependence (for pre-recruit survey) because most ecological logic places the density-dependent stage during the juvenile period following the larval stage that is most sensitive to environmental perturbation.
33	0	Sets expected survey abundance equal to age 0 recruitment.
34	0	Spawning biomass depletion (B/B_0)

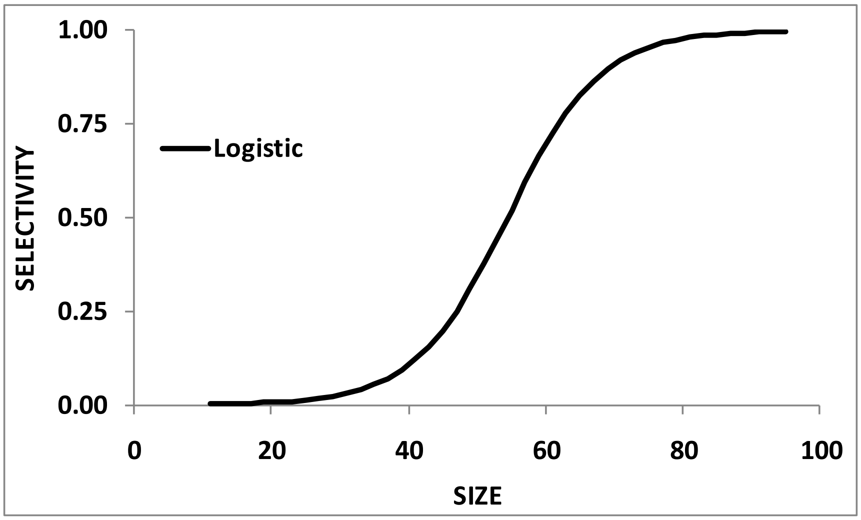
Do not input any size/age comp for surveys using pattern 30-33.
The "catchability" coefficient for these selectivity patterns 30-33 have all the general properties of the catchability coefficient for real surveys, e.g. they can be time-varying, use power relationship, etc.

AGE SELECTIVITY		
Pattern	N Parameters	Description
10	0	Age selex=1.0 for all ages beginning at age 1. If it is desired that age 0 fish be selected, then use pattern #11 and set the minimum age to 0.1
11	2	Pick min-max age
12	2	Logistic
13	8	Double logistic, IF joiners. Discouraged. Use pattern #18
14	nages+1	Each age, value at age is 1/(1+exp(-x))
15	0	Mirror another selex
16	2	Coleraine single Gaussian
17	Nages+1	Each age as random walk from previous age (New with version 3.0). For all ages in the population beginning with A_{min} = 1 for the fishery and 2 for the survey, there is a corresponding set of selectivity parameters for each fleet, p_a. The selectivity at age a is computed as, $$S_a = \exp\left(S'_a - S'_{\max}\right)$$ where S'_a is the sum of parameters for ages up to a, $$S'_a = \sum_{i=A_{\min}}^{a} p_a$$ and S'_{max} is the maximum of the S'_a, $$S'_{\max} = \max\left(S'_a\right)$$ Selectivity is fixed at $S_a = 0$ for $a < A_{min}$. This formulation has the properties that the maximum selectivity is equal to 1, positive p_a values are associated with increasing selectivity between ages a-1 and a, and negative values are associated with decreasing selectivity between those ages.
18	8	Double logistic, with defined peak, uses smooth joiners
19	6	Simple double logistic, no defined peak
20	6	Double normal with defined init and final level – Recommended option. Test using SELEX-24.xls
26	3	Exponential-logistic
27	3+2*N_nodes	Cubic Spline

9.3.19 Selectivity Details

Parameter usage (also see spreadsheet SS3-ExampleSetups.xls)

Pattern #1 – Logistic



Simple logistic function with two parameters.

p1 – size at inflection

p2 – width for 95% selection; a negative width causes a descending curve

Note that with a large p2 parameter, selectivity may not reach 1.0 at the largest size bin.

ADMB Code for logistic selectivity

```
if(seltype(f,3)<3 || (gg==1 && seltype(f,3)==3) || (gg==2 &&
seltype(f,3)==4))
   //  do the primary gender (gg)
   {sel = 1./(1.+mfexp(neglog19*(len_bins_m-sp(1))/sp(2)));}
   else  //  do the offset gender
   {
   temp=sp(1)+sp(Maleselparm(f));
   temp1=sp(2)+sp(Maleselparm(f)+1);
   sel = sp(Maleselparm(f)+2)/(1.+mfexp(neglog19*(len_bins_m-
temp)/temp1));
   }
```

Pattern 5 (mirror size) 2 parameters select the min and max Bin number (not min max size) of the source pattern. If first parameter has value <=0, then interpreted as value of 1 (e.g. first bin). If second parameter has value <=0, then interpreted as value of nlength (e.g. last bin). The source pattern must have a lower type number.

Pattern 6 (non-parametric size selectivity) uses a set of linear segments. The first waypoint is at Length = p1 and the last waypoint is at Length=p2. The total number of waypoints is specified by the value of the Special factor in the selectivity set-up, so the N intervals is one less than the number of waypoints. Intermediate waypoints are located at equidistant intervals between p1 and p2. Parameters 3 to N are the selectivity values at the waypoints, entered as logistic, e.g. 1/(1+exp(-x)). Ramps from –10 to p3 if L<p1. Constant at pN if L>p2. Note that prior to version 3.03 the waypoints were specified in terms of bin number, rather than length.

Patterns 1 (size) and 12 (age)

Simple logistic

Patterns 8 (size) and 18 (age)

Double logistic:

p1 – PEAK: size (age) for peak. Should be an integer and should be at bin boundary and not estimated. But options 7 and 18 may allow estimation

p2 – INIT: selectivity at lengthbin=1 (minL) or age=0

p3 – INFL1: size (age) at which selectivity is halfway between INIT and 1. A logit transform (1/(1+exp(-x)) is used so that the transformed value will be between 0 and 1. So a p1 value of –1.1 will be transformed to 0.25 and used to set the selectivity equal to 0.5 at a size (age) equal to 0.25 of the way between minL and PEAK. (see SS3-selex.xls).

p4 – SLOPE1: log(slope) of left side (ascending) selectivity

p5 – FINAL: logit transform for selectivity at maxL (or maxage)

p6 – INFL2: logit transform for size(age) at right side selectivity equal to half way between PEAK+PEAKWIDTH and maxL (or max age)

p7 – SLOPE2: log(slope) of right side (descending) selex

p8 – PEAKWIDTH: in width of flattop

Pattern 14 (age) Revise Age-selectivity pattern #14 to allow selectivity-at-age to be the same as selectivity at the next younger age. When using this option, the range on each parameter should be approximately -5 to 9 to prevent the parameters from drifting into extreme values with nil gradient.

```
case 14:                  // separate parm for each age
{
  temp=9.-max(sp(1,nages+1)); //forces at least one age to have
         selex weight equal to 9
    for (a=0;a<=nages;a++)
    {
       if(sp(a+1)>-999)
         {sel_a(y,f,1,a) = 1./(1.+mfexp(-(sp(a+1)+temp)));}
       else
         {sel_a(y,f,1,a) = sel_a(y,f,1,a-1);}
    }
}
```

Pattern 17 (age)

This selectivity pattern provides for a random walk in ln(selectivity). In typical usage:

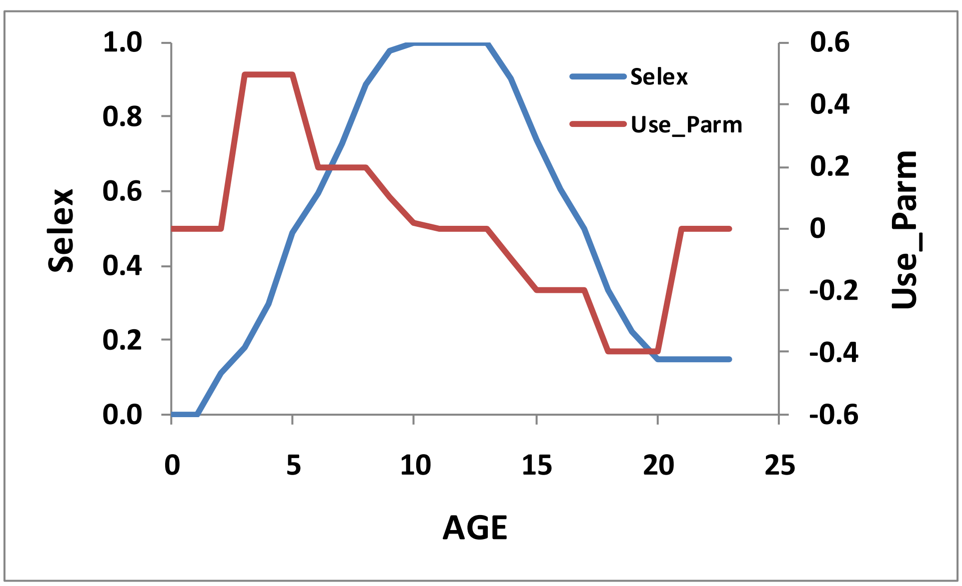
- First parameter (for age 0) could have a value of -1000 so that the age 0 fish would get a selectivity of 0.0;
- Second parameter (for age 1) could have a value of 0.0 and not be estimated, so age 1 is the reference age against which subsequent changes occur;

- Next parameters get estimated values. To assure that selectivity increases for the younger ages, the parameter min for these parameters could be set to 0.0 or a slightly negative value.
- If dome-shaped selectivity is expected, then the parameters for older ages could have a range with the max set to 0.0 so they cannot increase further.
- To keep selectivity at a particular age the same as selectivity at the next younger age, set its parameter value to 0.0 and not estimated. This allows for all older ages to have the same selectivity.
- To keep a constant rate of change in selectivity across a range of ages, use the -999 flag to keep the same rate of change in ln(selectivity) as for the previous age.

ADMB Code for Selectivity Pattern 17
(random walk in ln(selectivity)

```
case 17:              // separate parm for each age as random walk
{
   lastsel=0.0;  //  value is the change in log(selex); this is the reference value
for age 0
   tempvec=-999.;
   tempvec(0)=0.0;   //  so do not try to estimate the first value
   int lastage;
      if(seltype(f,4)==0)
          {lastage=nages;}
      else
          {lastage=abs(seltype(f,4));}

      for(a=1;a<=lastage;a++)
           {
             if(sp(a+1)>-999.) {lastsel=sp(a+1);}
               //so, lastsel stays constant until changed, so could create a linear
               change in ln(selex)
            // use of (a+1) is because the first element, sp(1), is  for age zero
             tempvec(a)=tempvec(a-1)+lastsel; //cumulative log(selex)
            }
    temp=max(tempvec);//find max so at least one age will have selex=1.
    sel_a(y,fs,1)=mfexp(tempvec-temp);
    a=0;
       while(sp(a+1)==-1000) //  reset range of young ages to selex=0.0
           {
             sel_a(y,fs,1,a)=0.0;
             a++;
           }
           if(lastage<nages)
           {
             for(a=lastage+1;a<=nages;a++)
             {
               if(seltype(f,4)>0)
               {sel_a(y,fs,1,a)=sel_a(y,fs,1,a-1);}
               else
               {sel_a(y,fs,1,a)=0.0;}
             }
           }
           break;
         }
```

			Max		
			2.22		
				Scaled_	
Age	Parm	Use_Parm	Cum_Parm	Cum_Parm	Selex
		0			
0	-1000	0.00	0.00	-2.22	0.000
1	-1000	0.00	0.00	-2.22	0.000
2	0	0.00	0.00	-2.22	0.109
3	0.5	0.50	0.50	-1.72	0.179
4	-999	0.50	1.00	-1.22	0.295
5	-999	0.50	1.50	-0.72	0.487
6	0.2	0.20	1.70	-0.52	0.595
7	0.2	0.20	1.90	-0.32	0.726
8	0.2	0.20	2.10	-0.12	0.887
9	0.1	0.10	2.20	-0.02	0.980
10	0.02	0.02	2.22	0.00	1.000
11	0	0.00	2.22	0.00	1.000
12	0	0.00	2.22	0.00	1.000
13	0	0.00	2.22	0.00	1.000
14	-0.1	-0.10	2.12	-0.10	0.905
15	-0.2	-0.20	1.92	-0.30	0.741
16	-0.2	-0.20	1.72	-0.50	0.607
17	-0.2	-0.20	1.52	-0.70	0.497
18	-0.4	-0.40	1.12	-1.10	0.333
19	-999	-0.40	0.72	-1.50	0.223
20	-999	-0.40	0.32	-1.90	0.150
21	0	0.00	0.32	-1.90	0.150
22	-999	0.00	0.32	-1.90	0.150
23	-999	0.00	0.32	-1.90	0.150



Pattern 9 (size) and 19 (age) – simple double logistic with no defined peak

p1 – INFL1: ascending inflection size (in cm)
p2 – SLOPE1: ascending slope
p3 – INFL2: descending inflection size (in cm)
p4 – SLOPE2: descending slope
p5 – first BIN: bin number for the first bin with non-zero selectivity (must be an integer bin number, not a size)
p6 – offset: enter 0 if P3 is independent of P1; enter 1 if P3 is an offset from P1

Pattern 22 – double normal with plateau

p1 – PEAK1: beginning size for the plateau (in cm)
p2 – PEAK2: ending size for the plateau. Calculated as a fraction of the distance between PEAK1 and 99% of the lower edge of the last size bin in the model. Transformed as (1/(1+exp(-p2)). So a value of 0 results in PEAK2 being halfway between PEAK1 and 99% of the last bin
p3 – upslope: ln(variance) on ascending side
p4 – downslope: ln(variance) on descending side

Pattern 23 and 24 (recommended double normal).
See spreadsheet SELEX-24.xls
p1 – PEAK: beginning size for the plateau (in cm)
p2 – TOP: width of plateau, as logistic between PEAK and MAXLEN
p3 – ASC-WIDTH: parameter value is ln(width)
p4 – DESC-WIDTH: parameter value is ln(width)
p5 – INIT: selectivity at first bin, as logistic between 0 and 1.
p6 – FINAL: selectivity at last bin, as logistic between 0 and 1. (for pattern #24)
or
p6 – FINAL: selectivity at last bin, as absolute value, so can be >1.0. (for pattern #23). Warning: Do not allow this value to go above 1.0 if the F_method uses Pope's approximation. OK to go above 1.0 when F is in exponential form. When this parameter is above 1.0, the overall selectivity pattern will have an intermediate plateau at 1.0 (according to peak and top), then will ascend further to the final value.

With SS_v3's separation of the population bin structure from the data bin structure, the interpretation of parameter p5 needed to change. Now, p5 refers to selex at the first DATA size bin and selex declines below that size according to (L/Lref)^2. Other recent changes include:

For the initial selectivity parameter (#5)

-999 or –1000: ignore the initial selectivity algorithm and simply decay the small fish selectivity according to P3,

< -1000: ignore the initial selectivity algorithm as above and then set selectivity equal to 1.0e-06 for size bins 1 through bin = -1001 –value. So a value of –1003 would set selectivity to a nil level for bins 1 through 2 and begin using the modeled selectivity in bin 3.

For the final selectivity parameter (#6),
-999 or –1000: ignore the final selectivity algorithm and simply decay the large fish selectivity according to parameter #4,
<-1000: set selectivity constant for bins greater than bin number = -1000 – value.

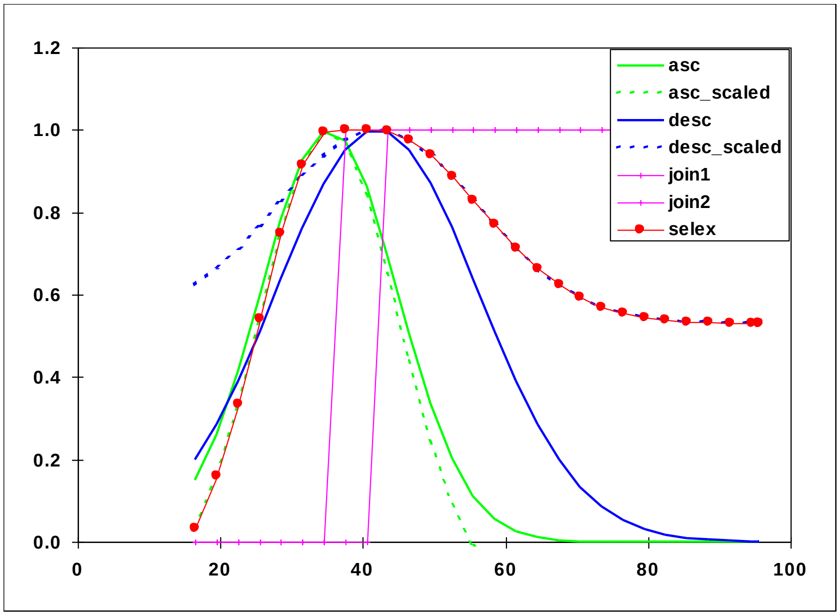


Figure 1 Selectivity pattern 24, double normal, showing sub-functions and steep logistic joiners

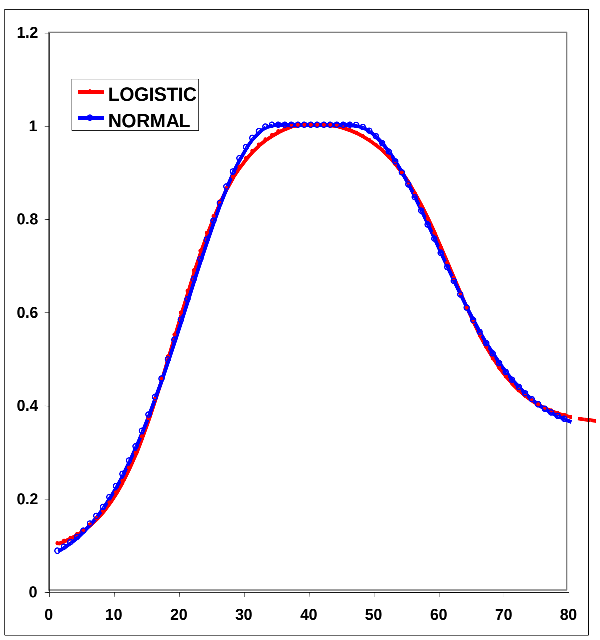


Figure 2. Comparison of 6-parameter double normal with 8-parameter double logistic tuned to match the double normal.

Pattern 15 (mirror age) no parameters. Whole age range is mirrored.
Pattern 16 Gaussian: like Coleraine.
p1 – age below which selectivity declines

p2 – scaling factor for decline

Pattern 9 and 19 (simple double logistic)

p1 – ascending inflection age/size

p2 – ascending slope

p3 – descending inflection age/size

p4 – descending slope

p5 – age or size at first selection; this is a specification parameter, so must not be estimated. Enter integer that is age for pattern 19 and is bin number for pattern 9

p6 – (0/1) where a value of 0 causes the descending inflection to be a standalone parameter, and a value of 1 causes the descending inflection to be interpreted as an offset from the ascending inflection. This is a specification parameter, so must not be estimated.

A value of 1.0e-6 is added to the selectivity for all ages, even those below the minage.

Pattern 25 (size) and 26 (age): Exponential-logistic

p1 – ascending rate, min: 0.02, max: 1.0, reasonable start value: 0.1

p2 – peak, as fraction of way between min size and max size. Parameter min value: 0.01; max: 0.99; reasonable start value: 0.5

p2 – minsize + p2*(maxsize-minsize)

p3 – descending rate, min: 0.001, max: 0.5, reasonable start value: 0.01. A value of 0.001 provides a nearly asymptotic curve. Values above 0.2 provide strongly dome-shaped function in which the p3 and p1 parameters interact strongly.

$$\frac{e^{p3*p1\left(p2'-size\right)}}{1-p3\left(1-e^{p1\left(p2'-size\right)}\right)}$$

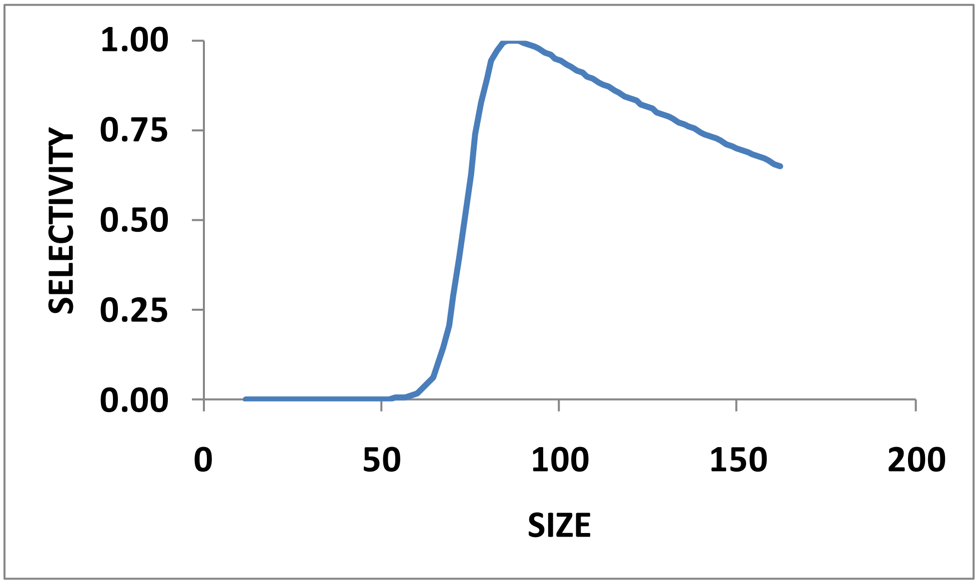


Figure 3. Exponential-logistic selectivity with p1=0.3, p2=0.5, p3=0. 02.

Pattern #27 Cubic Spline (age and size) -

This selectivity pattern uses the ADMB implementation of the cubic spline function. This function requires input of the number of nodes, the positions of those nodes, the parameter values at those nodes, and the slope of the function at the first and last node. In SS, the number of nodes is specified in the "special" column of the selectivity set-up. The pattern number 27 is used to invoke cubic spline for size selectivity and for age selectivity; the input syntax is identical.

For a 3 node setup, the SS input parameters would be:

p1 – code for initial set-up (0, 1 or 2) as explained below
p2 – gradient at the first node (should be a small positive value)
p3 – gradient at the last node (should be zero or a small negative value)
p4-p6 – the nodes in units of cm; must be in rank order and inside of the range of the population length bins. These must be held constant (not estimated, e.g. negative phase value) during a model run.
p7-p9 – the values at the nodes. Units are ln(selectivity).

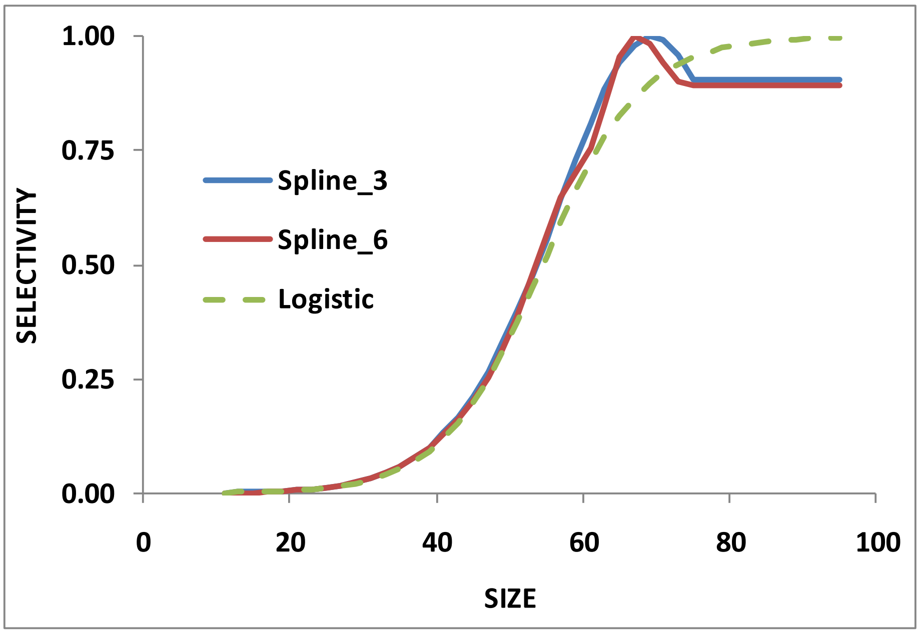
Notes:

- There must be at least 3 nodes.
- One of these selectivity parameter values should be held constant so others are estimated relative to it.
- Selectivity is forced to be constant for sizes greater than the size at the last node
- The overall selectivity curve is scaled to have a peak equal to 1.0.
- Terminal nodes cannot be at the min or max population length bins.

ADMB Code for cubic spline in SS

```
k=seltype(f,4);  // N nodes

for(i=1;i<=k;i++)
{
    splineX(i)=value(sp(i+3));  //  Nodes:    "value"  required  to  avoid
         error, but values should be always fixed anyway
    splineY(i)=sp(i+3+k);  //  selex parameter at each node
}

z=nlength;
while(len_bins_m(z)>splineX(k)) {z--;}
  j2=z+1;   //   first  size  bin  beyond  last  node  vcubic_spline_function
splinefn= vcubic_spline_function(splineX(1,k),splineY(1,k),sp(2),sp(3));
  tempvec_l = splinefn(len_bins_m);// interpolate selectivity at the mid-
      point of each population size bin
  temp=max(tempvec_l(1,j2));
  tempvec_l-=temp;  // rescale to get max of 0.0
  tempvec_l(j2+1,nlength) = tempvec_l(j2); //set constant above last node
  sel = mfexp(tempvec_l);
```

The figure below compares a 3 node and a 6 node cubic spline with a 2 parameter logistic function. In fitting these functions, the 2 cubic spline approaches fit slightly better than the logistic, presumably because the data were slightly indicative of a small dome in selectivity.



Auto-Generation of Cubic Spline Control File Set-Up

A New SS feature pioneered with the cubic spline function is a capability to produce more specific parameter labels and to auto-generate selectivity parameter setup. The auto-generation feature is controlled by the first selectivity parameter value for each fleet that is specified to use the cubic spline. There are 3 possible values for this setup parameter:

- 0: no auto-generation, process parameter setup as read.
- 1: auto-generate the node locations based on the specified number of nodes and on the cumulative size distribution of the data for this fleet/survey.
- 2: auto-generate the nodes and also the min, max, prior, init, and phase for each parameter.

With either the auto-generate option #1 or #2, it still is necessary to include in the parameter file placeholder rows of values so that the init_matrix command can input the current number of values because all selectivity parameter lines are read as a single matrix dimensioned as N parameters x 14 columns. The read values of min, max, init, prior, prior type, prior stddev, and phase will be overwritten.

Cumulative size and age distribution is calculated for each fleet, summing across all samples and both genders. These distributions are output in *echoinput.sso* and in a new OVERALL_COMPS section of *report.sso*.

When the nodes are auto-generated, the first node is placed at the size corresponding to the 2.5% percentile of the cumulative size distribution, the last is placed at the 97.5% percentile of the size distribution, and the remainder are placed at equally spaced percentiles along the cumulative size distribution. These calculated node values are output into control.ss_new. So, the user could extract these nodes from control.ss_new, edit them to desired values, then, insert them into the input control file. Remember to turn off auto-generation in the revised control file.

When the complete auto-generation is selected, the control.ss_new would look like the table below.

#_LO	HI	INIT	PRIOR	PR_ TYPE	SD	PHASE	Env-var	Use_ dev	Dev_ minyr	Dev_ maxyr	Dev_ stdev	Block	Block_ Fxn	#Label
0	2	2.00	0	-1	0	-99	0	0	0	0	0	0	0	#SizeSpline_Code_Fishery1
-0.001	1	0.13	0	1	0.001	3	0	0	0	0	0	0	0	#SizeSpline_GradLo_Fishery1
-1	0.001	-0.03	0	0	0.001	3	0	0	0	0	0	0	0	#SizeSpline_GradHI_Fishery1
11	95	38.08	0	-1	0	-99	0	0	0	0	0	0	0	#SizeSpline_Knot1_Fishery1
11	95	59.16	0	-1	0	-99	0	0	0	0	0	0	0	#SizeSpline_Knot2_Fishery1
11	95	74.55	0	-1	0	-99	0	0	0	0	0	0	0	#SizeSpline_Knot3_Fishery1
-9	7	-3.11	0	1	0.001	2	0	0	0	0	0	0	0	#SizeSpline_Val_1_Fishery1
-9	7	-1.00	0	-1	0	-99	0	0	0	0	0	0	0	#SizeSpline_Val_1_Fishery1
-9	7	-0.78	0	1	0.001	2	0	0	0	0	0	0	0	#SizeSpline_Val_1_Fishery1

Survey pattern #34 - depletion

This option allows a specified degree of stock depletion (in terms of spawning biomass) to be entered as the ratio of current year's spawning biomass relative to Bzero. With this option, it is not necessaryor reasonable to estimate the Q for this fleet, but you must set ln(Q) = 0.0 as a fixed value for absolute abundance. Also, if this option is used, then automatic adjustments to phase and lambda are made such that:

1. all parameter phases are adjusted by +1 so that only R0 is active in phase 1
2. all lambdas are set to 0 in phase 1, except the lambda for this depletion survey. Internally, the flag "depletion_fleet" is turned on (= to the index of that fleet) if there is a fleet with selex = #34

Essentially, these automated features cause SS to mimic DB-SRA in phase 1. If the model is only run through phase 1, then this will be the final result. Alternatively, use of this option could just be used to get the R0 parameter into a reasonable range before proceeding to estimate other parameters. The lambda for the depletion survey could remain at 1.0 for the entire model run, or it could be reduced in later phases to prevent influencing final model results.

9.3.20 Retention

Retention is defined as a logistic function of size. It does not apply to surveys. Four parameters are used:

- p1 – inflection
- p2 – slope
- p3 – asymptotic retention (often a time-varying quantity to match the observed amount of discard)
- p4 – male offset to inflection (arithmetic, not multiplicative)

$$\text{Retention} = \frac{P3}{1 + e^{-\frac{(L-(P1+P4*male))}{P2}}}$$

9.3.21 Discard mortality

Discard mortality is defined as a logistic function of size such that mortality declines from 1.0 to an asymptotic level as fish get larger. It does not apply to surveys and it does not affect the calculation of expected values for discard data. It is applied so that the total mortality rate is:

deadfish = selex * (retain + (1.0-retain)*discmort).

If discmort is 1.0, all selected fish are dead;

If discmort is 0.0, only the retained fish are dead.

Four parameters are used:

- p1 – inflection
- p2 – slope
- p3 – asymptotic mortality
- p4 – male offset to inflection (arithmetic, not multiplicative)

$$\text{Mortality} = 1 - \frac{(1-P3)}{\left(1 + \mathrm{e}^{-\frac{(L-(P1+P4*male))}{P2}}\right)}$$

9.3.22 Male Selectivity

There are two approaches to specifying gender specific selectivity. One approach allows male selectivity to be specified as a fraction of female selectivity (or vice versa). This first approach can be used for any selectivity pattern. The other option allows for separate selectivity parameters for each gender plus an additional parameter to define the scaling of one gender's peak selectivity relative to the other gender's peak. This second approach has only been implemented for a few selectivity patterns.

Approach #1:

If the "domale" flag is set to 1, then the selectivity parameters define female selectivity and the offset defined below sets male selectivity relative to female selectivity. The two genders switch roles if the "domale" flag is set to 2. Generally it is best to select the option so that the dependent gender has lower selectivity, thus obviating the need to rescale for selectivities that are greater than 1.0. Gender specific selectivity is done the same way for all size and age selectivity options.

P1 – size (age) at which a dogleg occurs (set to an integer at a bin boundary and do not estimate)

P2 – log(relative selectivity) at minL or age=0. Typically this will be set to a value of 0.0 (for no offset) and not estimated. It would be a rare circumstance in which the youngest/smallest fish had gender-specific selectivity.

P3 – log(relative selectivity) at the dogleg

P4 – log(relative selectivity) at maxL or max age.

For intermediate ages, the log values are linearly interpolated on size (age).

If selectivity for the dependent gender is greater than the selectivity for the first gender (which always peaks at 1.0), then the male-female selectivity matrix is rescaled to have a maximum of 1.0.

Approach #2:

A new gender selectivity option (3 or 4) has been implemented for size selectivity patterns 1 (logistic) and 23 and 24 (double normal) or age selectivity pattern 20 (double normal age). Rather than calculate male selectivity as an offset from female selectivity, here the male selectivity is calculated by making the male parameters an offset from the female parameters (option 3), or females are offset from males with option 4. The description below applies to option 3.

If the size selectivity pattern is 1 (logistic), then read 3 parameters

- male parm 1 is added to the first selectivity parm (inflection)
- male parm 2 is added to the second selectivity parm (width of curve)
- male parm 3 is the asymptotic selectivity

If the size selectivity pattern is 20, 23 or 24 (double normal), then

- male parm 1 is added to the first selectivity parm (peak)
- male parm 2 is added to the third selectivity parm (width of ascending side); then exp(this sum) per previous transform
- male parm 3 is added to the fourth selectivity parm (width of descending side); then exp(sum) per previous transform
- male parm 4 is added to the sixth selectivity parm (selectivity at final size bin); then 1/(1+exp(-sum)) per previous transform
- male parm 5 is the apical selectivity for males

Note that the male selectivity offsets currently cannot be time-varying (need to check on this). Because they are offsets from female selectivity, they inherit the time-varying characteristics of the female selectivity.

9.3.22 Reading the Selectivity and Retention Parameters

Read the required number of parameter setup lines as specified by the definition lines above. The complete order of the parameter setup lines is:

Size selectivity for fishery 1

Retention for fishery 1

Male offsets for size selectivity for fishery 1

<repeat for additional fleets and surveys>

Age selectivity for fishery 1

Male offsets for age selectivity for fishery 1

<repeat for additional fleets and surveys>.

The time-varying options for selectivity parameters are identical to the time-varying options for biology parameters. These options are described below in the Time-Varying Parameter Options section. After reading the selectivity parameters, which will include possible instructions to create environmental link, blocks, or dev vectors, then read the following section. Note that all inputs in this section are conditional (COND) on entries in the selectivity parameter section. So if no selectivity parameters invoke any time-varying properties, this section is left blank (or completely commented out with #).

VALUE		LABEL	DESCRIPTION
COND			If any selectivity parameters use environmental linkage, then read next line and associated parameter line(s)
	0	Custom_Env_linkage	
COND			If custom=0, then read one parameter line below and apply to all env fuctions; If custom>0, then read a setup line for each SEL-parm with Env-var>0. Note that control.ss_new will write out with custom=1 so it can write all the parameter values.
Enter proper number of short set-up lines (0, 1, several) for the SEL-parm environmental linkages. Each line will have 7 values: LO, HI, INIT, PRIOR, PR_type, SD, PHASE.			
COND			If any selectivity parameters use time blocks, then read next line and associated parameter line(s)
	0	Custom_block_setup	If custom=0, then read one setup and apply to all Block fxns; If custom>0, then read a setup line for each SEL-parm with Block>0.
Enter proper number of short set-up lines (0, 1, several) for the SEL-parm block linkages. Each line will have 7 values: LO, HI, INIT, PRIOR, PR_type, SD, PHASE.			
COND			If any selectivity parameters use annual devs, then read value
	-4	Selparm_dev_phase	Phase in which selparm devs, if any, are estimated
COND			If any selectivity parameters use environmental links, blocks or annual devs, then read value
	2	Selparm_Adjust_Method	1 = parameter adjustments for env, block and dev are applied directly and resulting value is not compared to base parameter bounds 2 = parameter adjustments use a logistic transformation to assure that adjusted parameter value stays within bounds of base parameter

9.3.23 Tag Recapture Parameters

Tag loss and Tag reporting parameters go next

VALUE	LABEL	DESCRIPTION
1	Tagging Data Present 0 = no read 1 = read following lines	

COND = 1

														LABEL
-10	10	9	9	1	0.001	-4	0	0	0	0	0	0	0	#TG_loss_init_1
-10	10	9	9	1	0.001	-4	0	0	0	0	0	0	0	#TG_loss_init_2
-10	10	9	9	1	0.001	-4	0	0	0	0	0	0	0	#TG_loss_init_3
-10	10	9	9	1	0.001	-4	0	0	0	0	0	0	0	#TG_loss_chronic_1
-10	10	9	9	1	0.001	-4	0	0	0	0	0	0	0	#TG_loss_chronic_2
-10	10	9	9	1	0.001	-4	0	0	0	0	0	0	0	#TG_loss_chronic_3
1	10	2	2	1	0.001	-4	0	0	0	0	0	0	0	#TG_overdispersion_1
1	10	2	2	1	0.001	-4	0	0	0	0	0	0	0	#TG_overdispersion_2
1	10	2	2	1	0.001	-4	0	0	0	0	0	0	0	#TG_overdispersion_3
-10	10	9	9	1	0.001	-4	0	0	0	0	0	0	0	#TG_report_fleet_1
-10	10	9	9	1	0.001	-4	0	0	0	0	0	0	0	#TG_report_fleet_2
-4	0	0	0	0	2	-4	0	0	0	0	0	0	0	#TG_rpt_decay_1
-4	0	0	0	0	2	-4	0	0	0	0	0	0	0	#TG_rpt_decay_2

9.3.24 Variance Adjustment Factors

When doing iterative reweighting of the input variance factors, it is convenient to do this in the control file, rather than the data file. This section creates that capability.

1	(0/1) Variance Adjustment Factors

There are six rows and a value for each Fleet&survey on each row.

Fleet/Survey 1	Fleet/Survey 2	Fleet/Survey 3	Fleet/Survey 4	
0	0	0	0	# Survey CV
0	0	0	0	# Discard stddev
0	0	0	0	# Mean bodywt stddev
1	1	1	1	# Length comp
1	1	1	1	# Age comp
1	1	1	1	# Size-at-age

Survey CV: The survey input variance (labeled survey CV) is actually the standard deviation of the ln(survey). The variance adjustment is added directly to this standard deviation. Set to 0.0 for no effect. Negative values are OK, but will crash if adjusted value becomes negative.

Discard: The input variance is the CV of the observation. Because this will cause observations of near zero discard to appear overly precise, the variance adjustment is added to the discard standard deviation, not to the CV. Set to 0.0 for no effect.

Mean body wt input variance is in terms of the CV of the observation: Because such data are typically not very noisy, the variance adjustment is added to the CV and then multipled by the observation to get the adjusted standard deviation of the observation.

Length composition input variance is in terms of an effective sample size: The variance adjustment is multipled times this sample size. Set variance adjustment to 1.0 for no effect.

Age composition is treated the same way as length composition:

Size-at-age input variance is the sample size for the N observations at each age: The variance adjustment is multiplied times these N values. Set to 1.0 for no effect.

Usage note: the report.sso output file contains information useful for determining if an adjustment of these input values is warranted to better match the scale of the average residual to the input variance scale.

Usage note: because the actual input variance factors are modified, it is these modified variance factors that are used when creating parametric bootstrap data files. So, the control files used to analyze bootstrap generated data files should have the variance adjustment factors reset to null levels.

9.3.25 Lambdas (emphasis factors)

These values are multiplied by the corresponding likelihood component to calculate the overall negative log likelihood to be minimized.

4	Max_lambda_phase: read this number of lambda values for each element below. The last lambda value is used for all higher numbered phases
1	sd_offset; value=0 causes log(like) to omit the +log(s) term; value=1 causes log(like) to include the log(s) term for CPUE, discard, meanbodywt, recruitment deviations.

USAGE Note: If the CV for size-at-age is being estimated and the model contains mean size-at-age data, then the flag for inclusion of the +log(stddev) term in the likelihood must be included. Otherwise, the model will always get a better fit to the mean size-at-age data by increasing the parameter for CV of size-at-age.

The reading of the lambda values has been substantially altered with SS_v3. Instead of reading a matrix containing all the needed lambda values, SS now just reads those elements that will be given a value other than 1.0. After reading the datafile, SS sets lambda equal to 0.0 if there are no data for a particular fleet/data type, and a value of 1.0 if data exist. So beware if your data files had data but you had set the lambda to 0.0 in a previous version of SS. First read an integer for the number of changes.

3	#number of changes to make to default Lambdas (default value is 1.0)			
# Then read that number of lines containing the change info:				
#Component	fleet/survey	Phase	Lambda	Sizefreq method
1	2	2	1.5	1
4	2	2	10.	1
4	2	3	2.	1

The codes for component are: 1=surv; 2=disc; 3=mnwt; 4=length; 5=age; 6=SizeFreq; 7=sizeage; 8=catch; 9=init_equ_catch; 10=recrdev; 11=parm_prior; 12=parm_dev; 13=CrashPen; 14=Morphcomp; 15=Tag-comp; 16=Tag-negbin.

On output to control.ss_new, the full table is written:

```
# lambdas (for info only; columns are phases)
#	0	0	0	0	#_CPUE/survey:_1
#	1	1.5	1.5	1.5	#_CPUE/survey:_2
#	1	1	1	1	#_CPUE/survey:_3
#	1	1	1	1	#_lencomp:_1
#	1	10	2	2	#_lencomp:_2
#	0	0	0	0	#_lencomp:_3
#	1	1	1	1	#_agecomp:_1
#	1	1	1	1	#_agecomp:_2
#	0	0	0	0	#_agecomp:_3
#	1	1	1	1	#_size-age:_1
#	1	1	1	1	#_size-age:_2
#	0	0	0	0	#_size-age:_3
#	1	1	1	1	#_init_equ_catch
#	1	1	1	1	#_recruitments
#	1	1	1	1	#_parameter-priors
#	1	1	1	1	#_parameter-dev-vectors
	1	1	1	1	#_crashPenLambda
```

9.3.26 Controls for variance of derived quantities

Add option to get variance estimates for one selectivity pattern and for size-at-age. At the end of the control file, just before the "999", add:

1		(0/1) read specs for more stddev reporting							
COND > 0			If the above value is "0", then do not include any more entries. If the above value is "1", then read 9 more integers:						
	Fleet	len/age	year	N selex bins	Growth pattern	N growth ages	Area for N_at_age	Year for N_at_age	N ages to report
	1	1	-1	7	1	5	-1	-1	5

Where:

- FLEET: is the index of the fleet to be output. A value of zero causes there to be no selectivity variance output;
- LEN/AGE: enter "1" to select length selex or "2" to select age selectivity. There is no option to get the combined age selectivity that incorporates the size selectivity;
- YEAR: enter a value for the selected year, or enter -1 to get the selectivity in the end year
- N Selectivity bins: enter the number of bins for which selectivity will be output. This number controls the number of items to be read below, even if the FLEET is set to zero. In other words, the read occurs even if the effect of the read is disabled.
- GROWTH PATTERN: growth pattern is the number of the growth pattern to be output. Enter "0" to get no variance output for size-at-age. Note that in a multiple season model, SS will output the size-at-age for the last birthseason that gets any recruits within the year. Also, if growth parameters are not estimated, then stddev output of mean size-at-age is disabled.
- N growth bins: Number of ages for which size-at-age variance is requested. This number controls the number of items to be read below, even if the growth pattern selection is set to zero. In other words, the read occurs even if the effect of the read is disabled.
- Area: specifies the area for which output of numbers at age is requested. A value of 0 disables this output. A value of -1 requests that numbers-at-age be summed across all areas. In all cases, numbers-at-age is summed across all growth patterns and sub-morphs and output for each gender.
- Year: specifies the year for which numbers-at-age are output. A value of -1 requests output for year equal to endyear+1, hence the year that starts the forecast period.
- N numbers-at-age bins: as with the N growth bins.

If the number of selex bins to be output is >0, then read a vector of selex bin numbers. For size selex, these are population bin numbers. For age selex, they refer directly to age. Entering a negative value for the first bin causes SS to self-generate an evenly spaced set.

5 10 16 22 27 38 46	Vector with selex std bin picks (-1 in first bin to self-generate)

If the number of growth bins to be output is >0, then read a vector of ages to be output. Entering a negative value for the first bin causes SS to self-generate a set that begins at AFIX, ends at Nages, and is denser at younger ages.

1 2 14 26 40	vector with growth std bin picks (-1 in first bin to self-generate)

If the number of numbers-at-age bins to be output is >0, then read a vector of ages to be output. Entering a negative value for the first bin causes SS to self-generate a set that begins at 1, ends at Nages, and is denser at younger ages.

1 2 14 26 40	vector with growth std bin picks (-1 in first bin to self-generate)

So a complete input looks like:

1	# (0/1) read specs for more stddev reporting								
COND > 0									
	1	1	-1	5	1	5	1	-1	5
	5	16	27	38	46	# vector with selex std bin picks (-1 in first bin to self-generate)			
	1	2	14	26	40	# vector with growth std bin picks (-1 in first bin to self-generate)			
	1	2	14	26	40	# vector with N-at-age std bin picks (-1 in first bin to self-generate)			

End Control File

999 #_end-of-file

9.3.27 Time-Varying Parameter Options

Biology parameters and selectivity parameters can vary over time. There are four options for time-varying parameters: blocks, trends, environmental linkage, and annual devs.

Elements 8 through 14 of the full parameter lines are used to setup the time-varying properties. If any parameter of the biology section is made to be time-varying, then one or more conditional inputs at the end of the biology section (or end of the selectivity section) will need to be turned on, and one or more parameter lines will need to be inserted to contain the parameter linkages and offsets that have been selected. This is done separately for the block of biology parameters and then for the selectivity parameters.

With version SS v3, the options for time-varying parameters have been expanded to include more additive effects. This is because it is not logical for a parameter whose range spans 0.0 to have a time-varying effect defined in a multiplicative way. This is especially true for those parameters that are exponentiated as they are being used. For example, the parameters that define the allocation of recruitment among areas and seasons should be made time-varying only through an additive function.

The order in which time-varying effects are calculated is: first blocks or trends, then environmental effects, then annual devs.

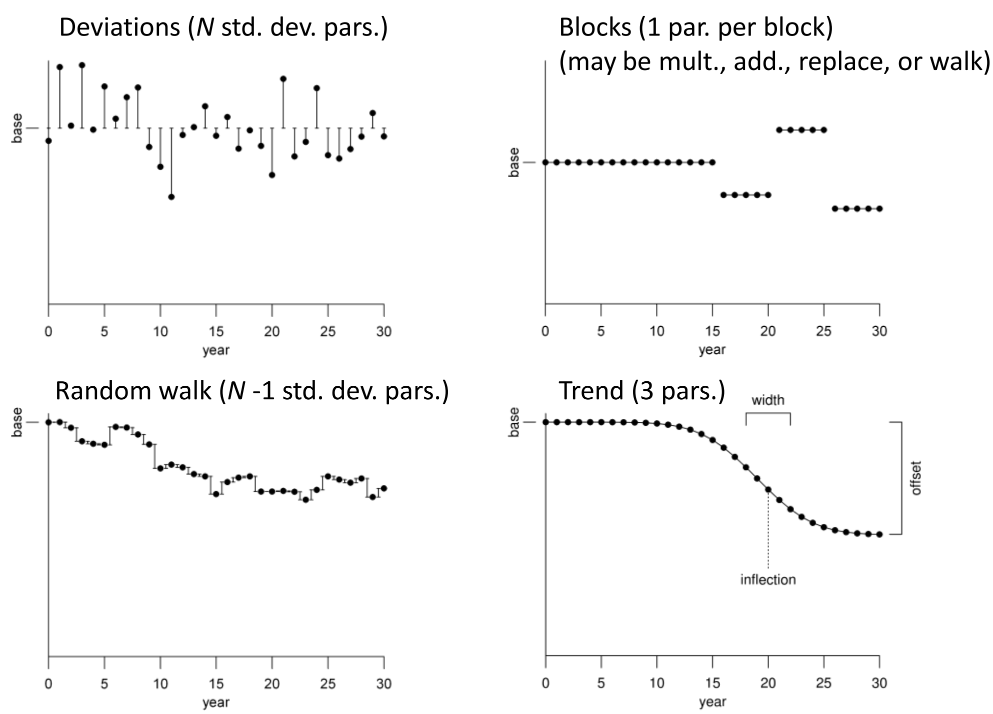
All time-varying options work on an annual time step, so in a multi-season model the parameters remain constant for the entire year. The exception is for the select biology parameters that have a separate capability to vary seasonally.

If the parameter adjustment method is set to a value of 2, then each parameter time-varying adjustments has an intermediate logistic transformation so the adjusted parameter stays within the min-max bounds of the parameter being adjusted. With this method, multiplicative adjustments are not implemented and the additive adjustments are in the domain of the logistic transformed base parameter. So, the adjustment coefficients will not have intuitive values.

The available options for time-varying parameters are described in the table below.

	Environ	Annual Devs				Blocks	
NAME:	Env Var	Use dev	Dev minyr	Dev maxyr	Dev stddev	Block	Block Fxn
ELEMENT:	8	9	10	11	12	13	14
OPTIONS:	>0: mult	1: mult	1973	1985	0.4	>0:block index	0: mult
	<0: additive	2: additive				<0: trend	1: additive
	abs(value): env index	3: additive randwalk					2: replace
							3: randwalk

Time-varying parameter options



also: Environmental link

ENV	A positive value, *g*, causes SS to set the annual working value of this parameter equal to a multiplicative function of Environmental Variable *g:parm'(y) = parm * exp(link * env(y,g))* A negative value, *g*, causes SS to set the annual working value of this parameter equal to a additive function of Environmental Variable *g:* *parm'(y) = parm + link * env(y,-g)* Where, *link* is the environmental link parameter, parm is the base parameter being adjusted, parm' is the value after adjustment, and *env(y,-g) is* the value of the environmental input g in year y. SS counts the number of parameters that invoke use of an Environmental Variable. After SS finishes reading the section's parameter lines, it then creates/reads additional short parameter line(s) to set up the link parameters. If custom=0, then one short parameter line is used to define the min, max, init, etc, for each of the link parameters. If custom=1, then a separate line is read for each.
USE_Dev	A value of 1 invokes multiplicative: parm'(y) = parm * exp(dev(y)) A value of 2 invokes additive: parm'(y) = parm + dev(y) A value of 3 invokes additive random walk: parm'(y) = parm'(y-1) + dev(y) The vector of devs is simply a vector of offsets; there is no inherent sum to zero constraint. However, the fact that they are each penalized by the DEV std.dev. below will tend to make them sum towards 0.0.
DEV min yr	Beginning year for the dev vector for this parameter
DEV max yr	Ending year for the dev vector for this parameter
DEV std.dev.	Standard deviation for elements in the dev vector for this parameter
USE-BLOCK	<u>Block</u>: A positive value identifies which block pattern will be used for time changes to a parameter value. Block patterns are simply numbered sequentially as they are defined near the top of the control file, so the index here must be correct for the order in which they are defined. More than one parameter can use the same block definition. The order of generated block parameters is by the order of the parameters that call for creation of the block parameters, then by the order of the blocks within that pattern. <u>Trend</u>: A negative value for the Use_Block input causes SS to create a parameter time trend instead of blocks. This time trend requires 3 parameters (instead of the normal one parameter per block). The base parameter is the value for the adjusted parameter in year = start year. For subsequent years, the three parameters define a normal distribution of change over time: P1: parameter value for year = end year. Either as logistic offset from base P (if Use_Block=-1), or as direct usage (if Use_Block=-2) P2: inflection year; if HI value for the base parameter is >1.1, then use as year, else use as fraction of range styr - endyr P3: width of change (units of std.dev. of years)
BLOCK-TYPE	This selects the way in which the block parameter creates an offset from the base parameter. 0 means that parm' = baseparm * exp(blockparm) 1 means that parm' = baseparm + blockparm 2 means that parm' = blockparm 3 means that parm' = is additive offset from previous block. Note that blocks must be contiguous to use this option properly.

9.3.28 Parameter Priors

Priors on parameters fulfill two roles in SS. First, for parameters provided with an informative prior, SS is receiving additional information about the true value of the parameter. This information works with the information in the data through the overall logL function to arrive at the final parameter estimate. Second, diffuse priors provide only weak information about the value of a prior and serve to manage model performance during execution. For example, some selectivity parameters may become unimportant depending upon the values of other parameters of that selectivity function. In the double normal selectivity function, the parameters controlling the width of the peak and the slope of the descending side become redundant if the parameter controlling the final selectivity moves to a value indicating asymptotic selectivity. The width and slope parameters would no longer have any effect on the logL, so they would have no gradient in the logL and would drift aimlessly. A diffuse prior would then steer them towards a central value and avoid them crashing into the bounds. Another benefit of diffuse priors is the control of parameters that are given unnaturally wide bounds. When a parameter is given too broad of a bound, then early in a model run it could drift into this tail and potentially get into a situation where the gradient with respect that parameter approaches zero even though it is not at its global best value. Here the diffuse prior helps move the parameter back towards the middle of its range where it presumably will be more influential and estimable. The options for parameter priors are:

<table>
<tr><th>PR_Type</th><th>PR_value, Pr</th><th>PR_stddev, Psd</th><th>Description</th></tr>
<tr><td>-1</td><td>NA</td><td>NA</td><td>No prior applied.</td></tr>
<tr><td>0</td><td>Pr</td><td>Psd</td><td>Normal prior. Note that this function is independent of the parameter bounds.</td></tr>
<tr><td colspan="4">Prior_Like = 0.5*square((Pval-Pr)/Psd);</td></tr>
<tr><td>1</td><td>NA</td><td>Psd</td><td>Symmetric beta prior is scaled between parameter bounds, so imposes larger penalty near bound. Psd=0.05 is very diffuse and a value of 5.0 provides a smooth U-shaped prior. See figure below.</td></tr>
<tr><td colspan="4">mu=-(Psd*(log((Pmax+Pmin)*0.5- Pmin)))- (Psd*(log(0.5)));
Prior_Like = -(mu+ (Psd*(log(Pval-Pmin+Pconst)))+
(Psd*(log(1.-((Pval-Pmin-Pconst)/(Pmax-Pmin))))));</td></tr>
<tr><td>2</td><td>Pr</td><td>Psd</td><td>Beta prior</td></tr>
<tr><td colspan="4">mu=(Pr-Pmin) / (Pmax-Pmin); // CASAL's v
tau=(Pr-Pmin)*(Pmax-Pr)/square(Psd)-1.0;
Bprior=tau*mu; Aprior=tau*(1.0-mu); // CASAL's m and n
if(Bprior<=1.0 || Aprior <=1.0) {warning<<" bad Beta prior "<<Pval<<" "<<Pr<<endl;N_warn++;}
Prior_Like = (1.0-Bprior)*log(Pconst+Pval-Pmin) +
(1.0-Aprior)*log(Pconst+Pmax-Pval) –(1.0-Bprior)*log(Pconst+Pr-Pmin) –
(1.0-Aprior)*log(Pconst+Pmax-Pr);</td></tr>
<tr><td>3</td><td>Pr</td><td>Psd</td><td>Lognormal prior. Note that lower bound on the parameter must be >0.0.</td></tr>
</table>

```
Prior_Like = 0.5*square((log(Pval)-Pr)/Psd);
```

where:

- Pval is value of the parameter for which a prior is being calculated;
- Pmin and Pmax are the bounds on the parameter;
- Pr is the value of the parameter prior, or the first of 2 factors controlling the calculation of the prior;
- Psd is the value of the prior's standard deviation, or the second of 2 factors controlling the calculation of the prior;
- Pconst is a small constant, 0.0001;
- Prior_Like is the calculated value of the prior's contribution to the logL.

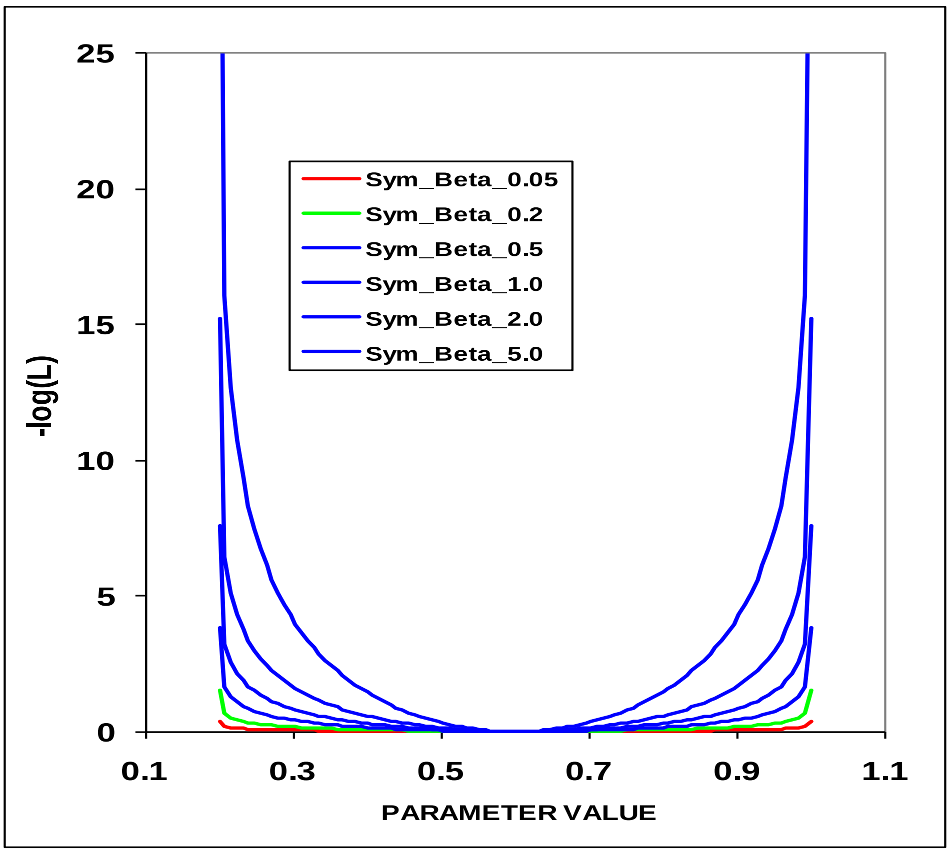


Figure 4. Prior distributions for the symmetric beta distribution.

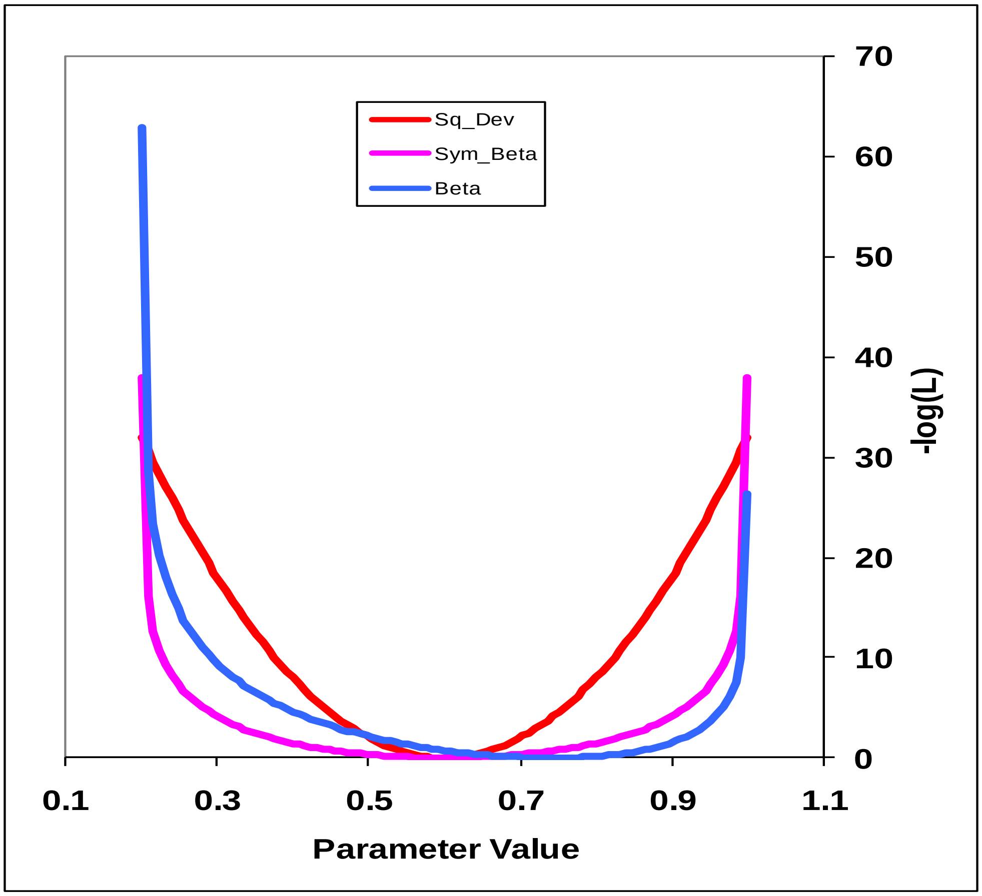


Figure 5 Comparison of parameter prior functions.

10. Output Files

10.1 Standard ADMB output files

Standard ADMB files are created by SS. These are:

SS3.PAR – This file has the final parameter values. They are listed in the order they are declared in SS. This file can be read back into SS to restart a run with these values (see running SS).

SS3.STD – This file has the parameter values and their estimated standard deviation for those parameters that were active during the model run. It also contains the derived quantites declared as sdreport variables. All of this information is also report in the covar.sso. Also, the parameter section of report.sso lists all the parameters with their SS generated names, denotes which were active in the reported run, displays the parameter standard deviations, then displays the derived quantities with their standard deviations.

SS3.REP – This report file is created between phases so, unlike report.sso, will be created even if the Hessian fails. It does not contain as much output as shown in report.sso.

SS3.COR – This is the standard ADMB report for parameter and sdreport correlations. It is in matrix form and challenging to interpret. This same information is reported in covar.sso.

10.2 Derived Quantities

Before listing the derived quantities reported to the sdreport, there are a couple of topics that deserve further explanation.

10.2.1 Metric for Fishing Mortality

A generic single metric of annual fishing mortality is difficult to define in a generalized model that admits multiple areas, multiple biological cohorts, dome-shaped selectivity in size and age for each of many fleets. Several separate indices are provided and others could be calculated by a user from the detailed information in report.sso.

10.2.2 Equilibrium SPR

This index focuses on the effect of fishing on the spawning potential of the stock. It is calculated as the ratio of the equilibrium reproductive output per recruit that would occur with the current year's F intensities and biology, to the equilibrium reproductive output per recruit that would occur with the current year's biology and no fishing. Thus it internalizes all seasonality, movement, weird selectivity patterns, and other factors. Because this index moves in the opposite direction than F intensity itself, it is usually reported as 1-SPR. A benefit of this index is that it is a direct measure of common proxies used for F_{MSY}, such as $F_{40\%}$. A shortcoming of this index is that it does not directly demonstrate the fraction of the stock that is caught each year. The SPR value is also calculated in the benchmarks (see below). The derived quantities report shows an annual SPR statistic. The options, as specified in the starter.ss file, are:

0=skip;

1=(1-SPR)/(1-SPR_tgt);

2=(1-SPR)/(1-SPR_MSY);

3=(1-SPR)/(1-SPR_Btarget);

4=rawSPR

10.2.3 F std

This index provides a direct measure of fishing mortality. The options are:

0=skip;

1=exploitation(Bio);

2=exploitation(Num);

3=sum(Frates)

The exploitation rates are calculated as the ratio of the total annual catch (in either biomass or numbers as specified) to the summary biomass or summary numbers on Jan 1. The sum of the F rates is simply the sum of all the apical Fs. This makes sense if the F method is in terms of instantaneous F (not Pope's approximation) and if there are not fleets with widely different size/age at peak selectivity, and if there is no seasonality, and especially if there is only one area. In the derived quantities, there is an annual statistic that is the ratio of the can be annual F_std value to the corresponding benchmark statistic. The available options for the denominator are:

0=raw;

1=F/Fspr;

2=F/Fmsy ;

3=F/Fbtgt

10.2.4 F-at-Age

Because the annual F is so difficult to interpret as a sum of individual F components, an indirect calculation of F-at-age is reported at the end of the report.sso file. This section of the report calculates Z-at-age simply as $\ln(N_{a+1,t+1}/N_{a,t})$. This is done on an annual basis and summed over all areas. It is done once using the fishing intensities as estimated (to get Z), and once with the F intensities set to 0.0 to get *M*-at-age. This latter sequence also provides a measure of dynamic Bzero. The user can then subtract the table of *M*-at-age/year from the table of Z-at-age/year to get a table of F-at-age/year. From this apical F, average F over a range of ages, or other user-desired statistics could be calculated. Further work within SS with this table of values is anticipated.

10.2.5 MSY and other Benchhmark Items

The following quantities are included in the sdreport vector mgmt_quantities, so obtain estimates of variance. Some additional quantities can be found in the benchmarks section of the forecast_report.sso.

Benchmark Item	Description
SSB_Unfished	Unfished reproductive potential (SSB is commonly female mature spawning biomass)
TotBio_Unfished	Total age 0+ biomass on Jan 1
SmryBio_Unfished	Biomass for ages at or above the summary age on Jan 1
Recr_Unfished	Unfished recruitment
SSB_Btgt	SSB at user specified SSB target
SPR_Btgt	Spawner potential ratio (SPR) at F intensity that produces user specified SSB target
Fstd_Btgt	F statistic at F intensity that produces user specified SSB target
TotYield_Btgt	Total yield at F intensity that produces user specified SSB target
SSB_SPRtgt	SSB at user specified SPR target (but taking into account the spawner-recruitment relationship)
Fstd_SPRtgt	F intensity that produces user specified SPR target
TotYield_SPRtgt	Total yield at F intensity that produces user specified SPR target
SSB_MSY	SSB at F intensity that is associated with MSY; this F intensity may be directly calculated to produce MSY, or can be mapped to F_SPR or F_Btgt
SPR_MSY	Spawner potential ratio (SPR) at F intensity associated with MSY
Fstd_MSY	F statistic at F intensity associated with MSY
TotYield_MSY	Total yield (biomass) at MSY
RetYield_MSY	Retained yield (biomass) at MSY

10.3 Brief cumulative output

Cum_Report.sso: contains a brief version of the run output, which is appended to current content of file so results of several runs can be collected together. This is especially useful when a batch of runs is being processed. Unless this file is deleted, it will contain a cumulative record of all runs done in that subdirectory. The first column contains the run number.

10.4 Output for Rebuilder Package

Output filename is REBUILD.DAT

```
#Title  # various run summary outputs
SS#_default_rebuild.dat
# Number of sexes
2
# Age range to consider (minimum age; maximum age)
0 40
# Number of fleets
1
# First year of projection (Yinit)
2002
# First Year of rebuilding period (Ydecl)
1999
# Is the maximum age a plus-group (1=Yes;2=No)
1
#Generate future recruitments using historical recruitments (1) historical recruits/spawner (2)  or a stock-recruitment (3)
3
# Constant fishing mortality (1) or constant Catch (2) projections
1
# Fishing mortality based on SPR (1) or actual rate (2)
1
# Pre-specify the year of recovery (or -1) to ignore
-1
# Fecundity-at-age
# 0 1 2 3 4 5 6 7 8 9 10 <deleted values>
 0 0.00159008 0.0128943 <deleted values>
#female fecundity; weighted by N in year Y_init across morphs and areas
# Age specific selectivity and weight adjusted for discard and discard mortality
 #selex and wt for gender,fleet: 1 1
 0.114191 0.174077 0.245545 <deleted values>
 0.0248207 0.0707095 0.157683 <deleted values>
 #selex and wt for gender,fleet: 2 1
 0.114191 0.174077 0.245545 <deleted values>
 0.0248207 0.0707095 0.157683 <deleted values>
# M and current age-structure in year Yinit: 2002
# gender = 1
0.1 0.1 0.1 0.1 0.1 <deleted values>
1037.67 696.042 1468.49 <deleted values>
# gender = 2
0.1 0.1 0.1 0.1 <deleted values
1037.67 696.042 1468.49 <deleted values>
# Age-structure at Ydecl= 1999
```

902.589 668.071 1145.47 537.282 <deleted values>
902.589 668.071 1145.47 537.282 <deleted values>
Year for Tmin Age-structure (set to Ydecl by SS3)
1999
Number of simulations
1000
recruitment and biomass
Number of historical assessment years
93
Historical data
year recruitment spawner in B0 in R project in R/S project
1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 <deleted values> 2001 2002 #years (with first value representing R0)
8853.43 8658.22 8651.96 8645.41 8638.43 8630.75 <deleted values> 1594.53 2075.34 #recruits; first value is R0 (virgin)
63679.5 63679.5 63679.3 63678.3 63673.9 63661.6 <deleted values> 8614.18 7313.2 #spbio; first value is S0 (virgin)
1 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 <deleted values> 0 0 # in Bzero
0 1 1 1 1 1 1 <deleted values> 1 1 0 0 0 # in R project
0 1 1 1 1 1 1 <deleted values> 1 1 0 0 0 # in R/S project
Number of years with pre-specified catches
0
catches for years with pre-specified catches go next
Number of future recruitments to override
0
Process for overiding (-1 for average otherwise index in data list)
Which probability to product detailed results for (1=0.5; 2=0.6; etc.)
3
Steepness sigma-R Auto-correlation
0.371059 0.5 0
Target SPR rate (FMSY Proxy); manually change to SPR_MSY if not using SPR_target
0.5
Target SPR information: Use (1=Yes) and power
0 20
Discount rate (for cumulative catch)
0.1
Truncate the series when 0.4B0 is reached (1=Yes)
0
Set F to FMSY once 0.4B0 is reached (1=Yes)
0
Percentage of FMSY which defines Ftarget
0.75
Maximum possible F for projection (-1 to set to FMSY)
-1
Conduct MacCall transition policy (1=Yes)

```
0
# Defintion of recovery (1=now only;2=now or before)
2
# Results for rec probs by Tmax (1) or 0.5 prob for various Ttargets (2)
1
# Definition of the 40-10 rule
10 40
# Produce the risk-reward plots (1=Yes)
0
# Calculate coefficients of variation (1=Yes)
0
# Number of replicates to use
10
# Random number seed
-99004
# Conduct projections for multiple starting values (0=No;else yes)
0
# File with multiple parameter vectors
rebuild.ss3
# Number of parameter vectors: value is placeholder only, user needs to change it
1
# User-specific projection (1=Yes); Output replaced (1->9)
0  5  0  0.1
# Catches and Fs (Year; 1/2/3 (F or C or SPR); value); Final row is -1
2002 1 1
-1 -1 -1
# Split of Fs
2002 0.6
-1  99
# Time varying weight-at-age (1=Yes;0=No)
0
# File with time series of weight-at-age data
none
# Use bisection (0) or linear interpolation (1)
1
# Target Depletion
0.4
# Project with Historical recruitments when computing Tmin (1=Yes)
0
# CV of implementation error
0
```

10.5 Bootstrap data files

Data.ss_new: contains a user-specified number of datafiles, generated through a parametric bootstrap procedure, and written sequentially to this file. These can be parsed into individual data files and re-run with the model. The first output provides the unaltered input data file (with annotations added). The second provides the expected values for only the data elements used in the model run. The third and subsequent outputs provide parametric bootstraps around the expected values.

10.6 Updated control file

Control.ss_new: Updated version of the control file with final parameter values replacing the Init parameter values. Note that, at this time, the dev vectors are not written to this file.

10.7 Forecast and reference points

FORECAST-REPORT.sso: This file contains output of fishery reference points and forecasts. It is designed to meet the needs of the Pacific Fishery Management Council's Groundfish Fishery Management Plan, but it should be quite feasible to develop other regionally specific variants of this output.

The vector of forecast recruitment deviations is estimated during an additional model estimation phase. This vector includes any years after the end of the recrdev time series and before or at the endyear. When this vector starts before the ending year of the time series, then the estimates of these recruitments will be influenced by the data in these final years. This is problematic, because the original reason for not estimating these recruitments at the end of the time series was the poor signal/noise ratio in the available data. It is not that these data are worse than data from earlier in the time series, but the low amount of data accumulated for each cohort allows an individual datum to dominate the model's fit. Thus, an additional control is provided so that forecast recruitment deviations during these years can receive an extra weighting in order to counter-balance the influence of noisy data at the end of the time series.

An additional control is provided for the fraction of the log-bias adjustment to apply to the forecast recruitments. Recall that R is the expected mean level of recruitment for a particular year as specified by the spawner-recruitment curve and R' is the geometric mean recruitment level calculated by discounting R with the log-bias correction factor e-0.5s^2. Thus a lognormal distribution of recruitment deviations centered on R' will produce a mean level of recruitment equal to R. During the modeled time series, the virgin recruitment level and any recruitments prior to the first year of recruitment deviations are set at the level of R, and the lognormal recruitment deviations are centered on the R' level. For the forecast recruitments, the fraction control can be set to 1.0 so that 100% of the log-bias correction is applied and the forecast recruitment deviations will be based on the R' level. This is certainly the configuration to use when the model is in MCMC mode. Setting the fraction to 0.0 during maximum likelihood forecasts would center the recruitment deviations, which all have a value of 0.0 in ML mode, on R. Thus would provide a mean forecast that would be more comparable to the mean of the ensemble of forecasts produced in MCMC mode. Further work on this topic is underway.

Note:

1. Cohorts continue growing according to their specific growth parameters in the forecast period rather than staying static at the endyr values.
2. Environmental data entered for future years can be used to adjust expected recruitment levels. However, environmental data will not affect growth or selectivity parameters in the forecast.

The top of this file shows the search for Fspr and the search for Fmsy so the user can verify convergence. Note: if the STD file shows aberrant results, such as all the standard deviations being the same value for all recruitments, then check the Fmsy search for convergence.

The F_{MSY} can be calculated, or set equal to one of the other F reference points per the selection made in STARTER.SS.

The reference point output is shown in the table below:

Management_report						
Steepness_R0_S0	0.371	8853	63680			
+	(B_in_mT;_N_in_thousands)					
Element	Value	B_per_Recr	B_per_R0	B_Total	N_per_Recr	N_total
Recr_unfished(R0)	--	1.000	1.000	8853		
SPB_unfished(S0)	--	7.193	7.193	63680		
BIO_Smry_unfished	--	18.493	18.493	163727		
+	+	+	+	+		
SPR_target	0.500					
SPR_calc	0.500					
Fmult	0.261					
Exploit(Y/Bsmry)	0.052					
Recruit	--	--	0.265	2343		
SPBio	--	3.596	--	8426		
YPR_encountered	--	0.550	--	1289		
YPR_dead	--	0.550	--	1289	0.257	602
YPR_retain	--	0.550	--	1289		
Biomass_Smry	--	10.517	--	24641		
+	+	+	+	+		
Btarget_rel_to_S0	0.500					
Btgt_calc	0.500					
SPR	0.712					
Fmult	0.116					
Exploit(Y/Bsmry)	0.024					
Recruit	--	--	0.702	6218		
SPBio	--	5.120	--	31840		
YPR_encountered	--	0.335	--	2085		
YPR_dead	--	0.335	--	2085	0.141	880
YPR_retain	--	0.335	--	2085		
Biomass_Smry	--	13.947	--	86730		
+	+	+	+	+		
calculate_FMSY						
SPR	0.660					
Fmult	0.144					
Exploit(Y/Bsmry)	0.030					
Recruits	--	--	0.622	5506		
SPBio	--	4.750	--	26157		
SPBmsy/SPBzero(using_S0)	--	0.411	--	--		
SPBmsy/SPBzero(using_endyear_LifeHistory)	--	0.411	--	--		
MSY_for_optimize	--	0.391	--	2151		
MSY_encountered	--	0.391	--	2151		
MSY_dead	--	0.391	--	2151	0.168	927
MSY_retain	--	0.391	--	2151		
Biomass_Smry	--	13.124	--	72263		

The forecast is done once using the Target SPR and once using the adjustments specified in the 40:10 section of forecast.ss input. Each section contains a time series of seasonal biomass and catch, followed by a time series of population numbers-at-age for each morph.

Forecast_using_Fspr															
Allocation_Pattern_as_in_endyear															
Harvest_Rates_by_Season&Fleet_(equals_selected_forecast_Fmult_*_Allocation_pattern															
Season	fleet:1														
1	0.0802														
+															
Forecast_recruitments_use_this_fraction_of_logbias_adj_before_endyr+1:	1														
and_this_value_after_endyr:	0														
Extra_emphasis_on_forecast_recrdevs_before_endyr+1:	1														
N_forecast_yrs;_and_with_stddev:	6	6													
OY_Control:	Top;	Bottom;	Scale												
+	0.4	0.1	1												
+															
FORECAST:_Without_40:10															
pop	year	season	4010	bio-all	bio-Smry	SpawnBio	Depletion	recruit-0	dead_cat_B-1	retain_B-1	dead_cat_N-1	retain_N-1	Hrate-1	opt	ABC
1	2002	1	1	22361	21634	7313	0.115	2075	1116	1116	533	533	0.0802	R	1116
1	2003	1	1	22472	21613	7341	0.115	2082	1123	1123	534	534	0.0802	R	1123
+															
Forecast-NUMBERS_AT_AGE	<not shown here>														
FORECAST:__with_40:10_Adjustmen	0.4	0.1	1												
pop	year	season	4010	bio-all	bio-Smry	SpawnBio	Depletion	recruit-0	dead_cat_B-1	retain_B-1	dead_cat_N-1	retain_N-1	Hrate-1	opt	ABC
1	2002	1	0.17242	22361	21634	7313	0.115	2075	192	192	92	92	0.0138	R	1116
1	2003	1	0.226996	23357	22481	7674	0.121	2163	266	266	125	125	0.0182	R	1171
+															
Annual_all_area_values_stored_in_sdreport_vector_'depletion'_beginning_in_element_6															
Year	Spbio	Recruits	Depletion	Catch_or_	Exploitation										
2002	7313	2075	0.115	192.4	0.0089										
2003	7674	2163	0.121	265.8	0.0118										

where:

40:10	is the magnitude of the adjustment of harvest multiplier to implement the OY policy
bio-all	is the biomass of all ages
bio-smry	is the biomass for ages at or above the summary age
Spawnbio	Is the female spawning output
Depletion	is the spawnbio divided by the unfished spawnbio
Recruit-0	is the recruitment of age-o fish in this year
Dead_cat_B-1	is the total dead (retained plus dead discard) catch in MT for fleet 1
Retain_B-1	is fleet 1's retained catch in MT

Equivalent catch in numbers is then reported.

Hrate-1	is the harvest rate, as adjusted by the 40:10 policy. The units will depend on the F method selected (Pope's method giving mid-year harvest rate or the continuous F.
Opt=C	means that the rate was calculated from an input catch level (and crashed means that this caused an excessive harvest rate.
Opt=R	means that the catch was calculated from the target harvest rate.
ABC	is equal to the Total-Catch when the 40:10 option is not used (upper portion of table). When the 40:10 is on (lower table), the ABC is the catch level corresponding to no 40:10 adjustment after accounting for catch in previous year's from the 40:10.

The time series output described above is detailed by season, area, morph and fishery. It is usually more convenient to have annual values summed across areas, morphs and fisheries. This is done for the 40:10 output and a subset of these values are replicated in the depletion vector in the sd_report so that variance estimates can be obtained. The elements of the depletion vector in the sd_report are:

1. depletion level in end year
2. depletion level in end year+1
3. MSY (if calculated, else spbio in endyr-1)
4. BMSY (if calculated, else spbio in endyr)
5. SPRMSY (if calculated, else spbio in endyr+1) then the time series of:
    a. Spawning biomass
    b. Recruitment
    c. Depletion level
    d. Total catch (if forecast calculated catch from rates) or sum of fishery-specific harvest rates (if forecast is based on fixed input catch level in this year)
    e. Total exploitation rate (total dead catch divided by the summary biomass at the beginning of the year).

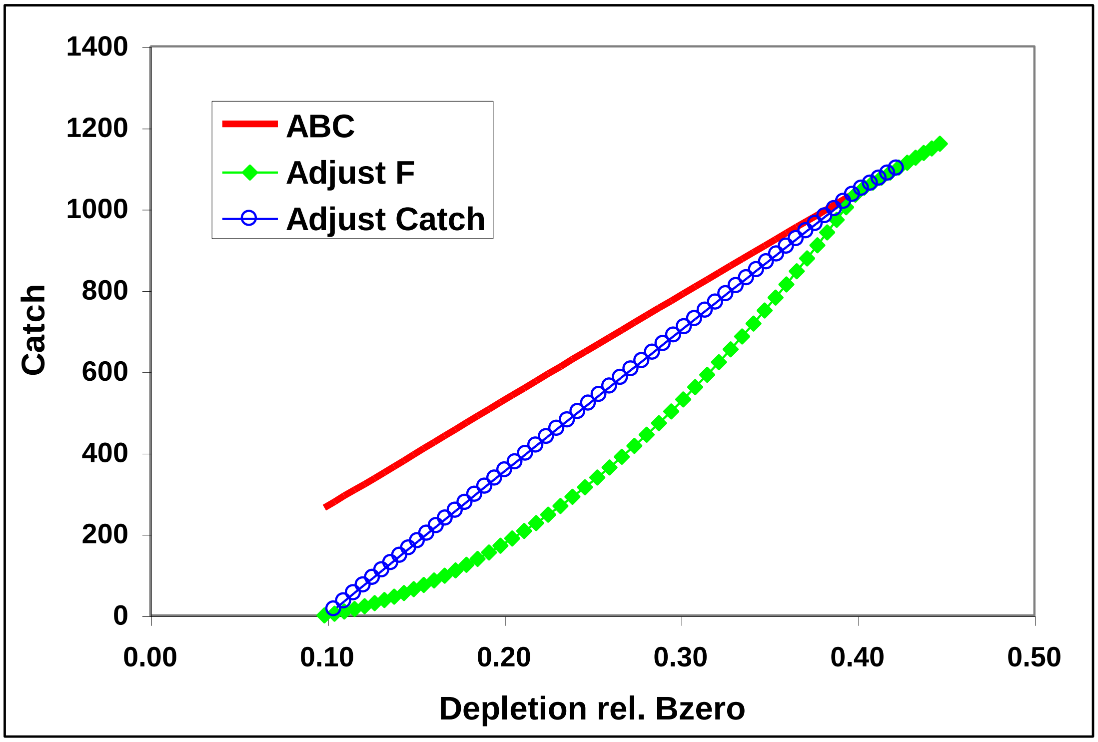


Figure 6. Two examples of harvest forecast adjustment: one adjusts catch and the other adjusts F.

10.8 Main output file, report.sso

Report.sso: This is the primary output file. Its major sections are listed below. Each has an associated label. The Excel spreadsheet ss3-output.xls (and ss3-output-x.xls for Office 2007) reads report.sso and compreport.sso, searches for labels in the first column, and copies the adjacent information into specific worksheets for detailed display. Similar capability has been built using R routines and has been included in the GUI.

The sections of the output file are:

1. SS version number with date compiled. Time and date of model run. This info appears at the top of all output files.
2. Comments
    a. Input file lines starting with #C are echoed here
3. KeyWords
    a. List of keywords used in searching for output sections.
4. FleetNames
    a. List of fishing fleet and survey names assigned in the data file
5. LIKELIHOOD
    a. Final values of the negative log(likelihood) are presented.
6. Input_Variance_Adjustment
    a. The matrix of input variance adjustments is output here because these values affect the logL calculations
7. PARAMETERS
    a. The parameters are listed here. For the estimated parameters, the display shows: Num (count of parameters), Label (as internally generated by SS), Value, Active_Cnt, Phase, Min, Max, Init, Prior, Prior_type, Prior_SD, Prior_Like, Parm_StD (standard deviation of parameter as calculated from inverse Hessian), Status (e.g. near bound), Pr_atMin (value of prior penalty if parameter was near bound), and Pr_atMin. The Active_Cnt entry is a count of the parameters in the same order they appear in the ss3.cor file.
8. Derived_Quantities

This section starts by showing the options selected from the starter.ss and forecast.ss input files:

SPR_ratio_basis:	(1-SPR)/(1-SPR_40%)
F_report_basis:	(F)/(Fmsy);_with_F=sum(full_Fs)
B_ratio_denominator:	40%*Virgin_Biomass

Then the time series of output, with standard deviation of estimates, are produced with internally generated labels. Note that these time series extend through the forecast era. The order of the output is: spawning biomass, recruitment, SPRratio, Fratio, Bratio, management quantities, forecast catch (as a target level), forecast catch as a limit level (OFL), Selex_std, Grow_std, NatAge_std. For the three "ratio" quantities, there is an additional column of output showing a Z-score calculation of the probability that the ratio differs from 1.0. The "management quantities" section is designed to meet the terms of reference for west coast groundfish assessments; other formats could be made available upon request. The _std quantities at the end are set up according to specifications at the end of the control input file. In some cases, a user may specify that no derived quantity output of a certain type be produced. In those cases, SS substitutes a repeat output of the virgin spawning biomass so that vectors of null length are not created.

ADMB NOTE: while vectors of null length are very useful for controlling optional model inputs, they cannot be used with current version of ADMB for sdreport quantities.

<u>MGparm_By_Year_after_adjustments</u>
This block shows the time series of Mgparms by year after adjustment by environmental links, blocks and deviations.

<u>SELparm(Size)_By_Year_after_adjustments</u>
This block shows the size selectivity parameters, after adjustment, for each year in which a change occurs.

<u>SELparm(Age)_By_Year_after_adjustments</u>
This block shows the age selectivity parameters, after adjustment, for each year in which a change occurs.

<u>RECRUITMENT_DIST</u>
This block shows the distribution of recruitment across growth patterns, genders, birthseasons, and areas in the endyr of the model.

<u>MORPH_INDEXING</u>
This block shows the internal index values for various quantities. It can be a useful reference for complex model setups. The vocabulary is: Bio_Pattern refers to a collection of cohorts with the same defined growth and natural mortality parameters; Gender is the next main index. If recruitment occurs in multiple seasons, then Birthseas is the index for that factor. The index labeled "Morph" is used as a continuous index across all the other factor-specific indices. If sub-morphs are used, they are nested within the Bio_Pattern x Gender x Birthseas morphs. However, some of the output tables use the column label "morph" as a continuous index across morphs and sub-morphs. Note that there is no index here for area. Each of the cohorts is distributed across areas and they retain their biological characteristics as they move among areas.

<u>SIZEFREQ_TRANSLATION</u>
If the generalize size frequency approach is used, this block shows the translation probabilities between population length bins and the units of the defined size frequency method. If the method uses body weight as the accumulator, then output is in corresponding units.

<u>MOVEMENT</u>
This block shows movement rate between areas in a multi-area model.

<u>EXPLOITATION</u>
This block shows the time series of the selected F_std unit and the F multiplier for each fleet in terms of harvest rate (if Pope's approximation is used) or fully selected F.

TIME_SERIES

This section reports the time series of abundance, recruitment and catch for each of the areas (populations). It extends into the forecast period. Output quantities include summary biomass and summary numbers for each gender and G_pattern. For each fishing fleet, the output includes: encountered (e.g. selected) catch biomass, dead catch biomass (which takes into account discard and discard mortality), retained catch biomass, same 3 quantities in terms of numbers, the observed catch (input catch value which should be closely matched by the retained catch) and the fully selected F multiplier. So even though input catch is either in terms of biomass or numbers, the output catch is shown in both.

The final column shows the spawning biomass as calculated with the start year's fecundity-at-age. If there are time-varying life history parameters, this column will show the impact of these changes on the calculation of spawning biomass in comparison to the spawning biomass calculated with the current year's life history.

SPR_series

This section reports on the yield per recruit and biomass per recruit calculations according to the current year's life history, fishery selectivity and fishing intensity. It is annual, so accumulates the effects across seasons and areas (if any are defined). The level of recruitment used for the calculations is R0, the virgin recruitment level. The report contains: Bio_all (fished total biomass at R0); Bio_Smry(fished summary biomass at R0); SPBzero (unfished spawning biomass at R0 and current year's life history); SPBfished (fished spawning biomass at R0 and current year's life history); SPBfished/R (fished spawning biomass per recruit); SPR (spawning potential ratio equal to SPBfished/SPBzero); Y/R (yield per recruit); GenTime (unfished generation time equal to mean age weighted by fecundity at age). Additional exploitation statistics are the AveF (average F across ages beginning at the summary age) for each gender, and the maxF among ages for each gender. Note that these two F quantities do not distinguish among areas in a multi-area model. This report section extends into the forecast period.

SPAWN_RECRUIT

This section shows the spawner output and recruitment. The column "exp-recr" shows the level estimated by the spawner-recruitment curve. The column "with-env" adjusts this value according to the inputted environmental conditions for that year. The column "bias-adj" shows the downward adjustment of the central tendency for the log-scale deviations so that the expected value for the mean recruitment will be unbiased. Finally, the column "pred-recr" shows the value used in the model after adjusting for the year specific recruitment deviation. Early years (prior to start of the recruitment deviations), including the virgin recruitment level, use the "exp-recr" level of recruitment. If the recruitment deviations stop before the ending year, then recruitment deviations for those years can be estimated as forecast recruitment deviations and will be labeled with "forecast" in this output. The root mean squared error among the recruitment deviations within the early and main eras is shown and compared to sigmaR and to the mean bias adjustment in order to guide the fine-tuning of the max_biasadjustment, the ramp in bias_adjustment, and the value of sigmaR.

INDEX_1

This section lists the options used for processing the abundance index data.

INDEX_2
This section reports the observed and expected values for each index. All are reported in one list with index number included as a selection field. At the bottom of this section, the root mean squared error of the fit to each index is compared to the mean input error level to assist the user in gauging the goodness-of-fit and potentially adjusting the input level of imprecision.

INDEX_3
This section shows the parameter number assigned to each parameter used in this section.

DISCARD
This is the list of observed and expected values for the amount (or fraction) discard.

MEAN_BODY_WT
This is the list of observed and expected values for the mean body weight.

FIT_LEN_COMPS
This is the list of the goodness of fit to the length compositions. The input and output levels of effective sample size are shown as a guide to adjusting the input levels to better match the model's ability to replicate these observations.

FIT_AGE_COMPS
This has the same format as the length composition section.

FIT_SIZE_COMPS
This has the same format as the length composition section and is used for the generalized size composition summary.

LEN_SELEX
Here is the length selectivity and other length specific quantities for each fishery and survey.

AGE_SELEX
Here is reported the time series of age selectivity and other age-related quantities for each fishery and survey. Some are directly computed in terms of age, and others are derived from the combination of a length-based factor and the distribution of size-at-age.

ENVIRONMENTAL_DATA
The input values of environmental data are echoed here. In the future, the summary biomass in the previous year will be mirrored into environmental column –2 and that the age zero recruitment deviation into environmental column –1. Once so mirrored, they may enable density-dependent effects on model parameters.

NUMBERS_AT_AGE
The output (in thousands of fish) is shown for each cohort tracked in the model.

NUMBERS_AT_LENGTH
The output is shown for each cohort tracked in the model.

CATCH_AT_AGE
The output is shown for each fleet. It is not necessary to show by area because each fleet operates in only one area.

BIOLOGY
The first biology section shows the length-specific quantities in the ending year of the time series only. The derived quantity spawn is the product of female body weight, maturity and fecundity per weight. The second section shows natural mortality.

GROWTH_PARAMETERS
This section shows the growth parameters, and associated derived quantities, for each year in which a change is estimated.

Biology_at_Age
This section shows derived size-at-age and other quantities. It is the basis for the Bio report page of the Excel output processor.

MEAN_BODY_WT(begin)
This section reports the time series of mean body weight for each morph. Values shown are for the beginning of each season of each year.

MEAN_SIZE_TIMESERIES
This section shows the time series of mean length-at-age for each morph. At the bottom is the average mean size as the weighted average across all morphs for each gender.

AGE_LENGTH_KEY
This is reported for the midpoint of each season in the ending year.

AGE_AGE_KEY
This is the calculated distribution of observed ages for each true age for each of the defined ageing keys.

Selectivity_Database
This section contains the selectivities organized as a database, rather than as a set of vectors.

Spawning_Biomass_Report_2, etc.
The section shows annual total spawning biomass, then numbers-at-age at the beginning of each year for each Bio_Pattern and Gender as summed over sub-morphs and areas. Then Z-at-age is reported simply as ln(Nt+1,a+1 / Nt,a). Then the Report_1 section loops back through the time series with all F values set to zero so that a dynamic Bzero, N-at-age, and M-at-age can be reported. The difference between Report_1 and Report_2 can be used to create an aggregate F-at-age.

Composition_Database
This section is reported to a separate file, compreport.sso, and contains the length composition, age composition, and mean size-at-age observed and expected values. It is arranged in a database format, rather than an array of vectors. Software to filter the output allows display of subsets of the database.

11. Running SS

SS can be run from a DOS window in a directory containing the file SS3.EXE (or a path to SS3.EXE) and the necessary input files. Simply type SS3 at the DOS prompt. This will start the executable and the first step will be to open and read the file named *starter.ss* which contains necessary run information.

As with all ADMB-based programs, switches are typed immediately after a space. For example, SS3 –nohess, would run SS3 without calculating the Hessian matrix.

11.1 Example of DOS batch input file

One file management approach is to put SS3.EXE in its own folder (example: C:\SS_model) and to put your input files in separate folder (example: C:\My Documents\SS_runs). Then a DOS batch file in the SS_runs folder can be run at the command line to start SS3.EXE. All output will appear in SS_runs folder.

A DOS batch file (e.g. SS.bat) might contain some explicit ADMB commands, some implicit commands, and some DOS commands:

```
c:\SS_model\ss3.exe -cbs 5000000000 -gbs 50000000000 %1 %2 %3 %4
del ss3.r0*
del ss3.p0*
del ss3.b0*
```

In this batch file, the –cbs and –gbs arguments allocate a large amount of memory for SS to use (you may need to edit these for your computer and SS configuration), and the %1, %2 etc. allows passing of command line arguments such as –nox or –nohess. You add more items to the list of % arguments as needed.

An easy way to start a command line in your current directory (SS_runs) is to create a shortcut to the DOS command line prompt. The shortcut's target would be:

%SystemRoot%\system32\cmd.exe

And it would start in %CURRDIR%

11.1.1 Simple batch

This first example relies upon having a set of prototype files that can be renamed to *starter.ss* and then used to direct the running of SS. The example also copies one of the output files to save it from being overwritten. This sequence is repeated 3 times here and can be repeated an unlimited number of times. The batch file can have any name with the .bat extension, and there is no particular limit to the DOS commands invoked. Note that brief output from each run will be appended to cumreport.sso (see below).

```
del ss3.cor
del ss3.std
copy starter.r01 starter.ss
c:\admodel\ss\ss3.exe -sdonly
copy ss3.std ss-std01.txt
copy starter.r01 starter.ss
c:\admodel\ss\ss3.exe -sdonly
copy ss3.std ss-std02.txt
```

11.1.2 Complicated batch

This second example processes 25 dat files from a different directory, each time using the same ctl and nam file. The loop index is used in the file names, and the output is searched for particular keywords to accumulate a few key results into the file *SUMMARY.TXT*. Comparable batch processing can be accomplished by using R or other script processing programs.

```
del summary.txt
del ss3-report.txt
copy /Y runnumber.zero runnumber.ss3
FOR /L %%i IN (1,1,25) DO (
copy /Y ..\MakeData\A1-D1-%%i.dat  Asel.dat
del ss3.std
del ss3.cor
del ss3.par
c:\admodel\ss\ss3.exe
copy /Y ss3.par A1-D1-A1-%%i.par
copy /Y ss3.std A1-D1-A1-%%i.std
find "Number" A1-D1-A1-%%i.par >> Summary.txt
find "hessian" ss3.cor >> Summary.txt)
```

11.1.3 Batch using PROFILEVALUES.SS

This example will run a profile on natural mortality and spawner-recruitment steepness, of course. Edit the control file so that the natural mortality parameter and steepness parameter lines have the phase set to –9999. Edit *STARTER.SS* to refer to this control file and the appropriate data file

Create a PROFILEVALUES.SS file

```
2     # number of parameters using profile feature
0.16  # value for first selected parameter when runnumber equals 1
0.35  # value for second selected parameter when runnumber equals 1
0.16  # value for first selected parameter when runnumber equals 2
0.40  # value for second selected parameter when runnumber equals 2
0.18  # value for first selected parameter when runnumber equals 3
0.40  # value for second selected parameter when runnumber equals 3
```

etc.; make it as long as you like.

Create a batch file that looks something like this. Or make it more complicated as in the example above.

```
del cumreport.sso
copy /Y runnumber.zero runnumber.ss        % so you will start with runnumber=0
```

C:\SS3\ss3.exe

C:\SS3\ss3.exe

C:\SS3\ss3.exe

C:\SS3\ss3.exe

Repeat as many times as you have set up conditions in the PROFILEVALUES.SS file.

The summary results will all be collected in the cumreport.sso file. Each step of the profile will have an unique Runnumber and its output will include the values of the natmort and steepness parameters for that run.

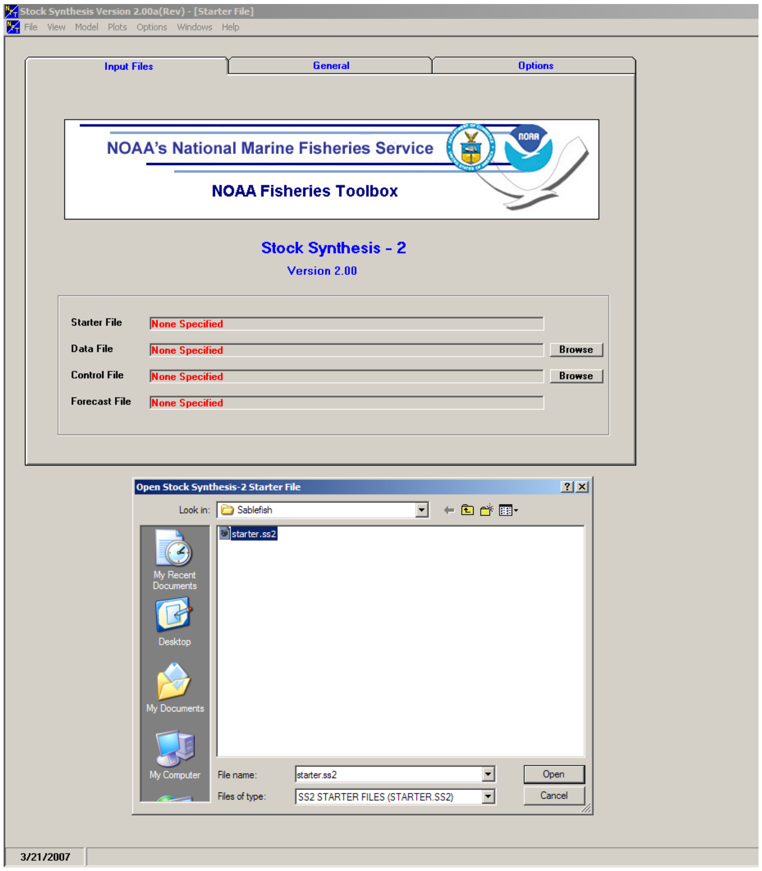
11.2 Re-starting a Run

SS model runs can be restarted from a previously estimated set of parameter values. In the *starter.ss* file, enter a value of 1 on the first numeric input line. This will cause SS to read the file *ss3.par* and use these parameter values in place of the initial values in the control file. This option only works if the number of parameters to be estimated in the new run is the same as the number of parameters in the previous run because only actively estimated parameters are saved to the file *ss3.par*. The file *ss3.par* can be edited with a text editor, so values can be changed and rows can be added or deleted. However, if the resulting number of elements does not match the setup in the control file, then unpredictable results will occur. Because *ss3.par* is a text file, the values stored in it will not give exactly the same initial results as the run just completed. To achieve greater numerical accuracy, SS can also restart from *ss3.bar* which is the binary version of *ss3.par*. In order to do this, the user must make the change described above to the *starter.ss* file and must also enter **–binp ss3.bar** as one of the command line options.

11.3 Graphical Interface

<Note: This section on the GUI has not been updated for SS_v3.>

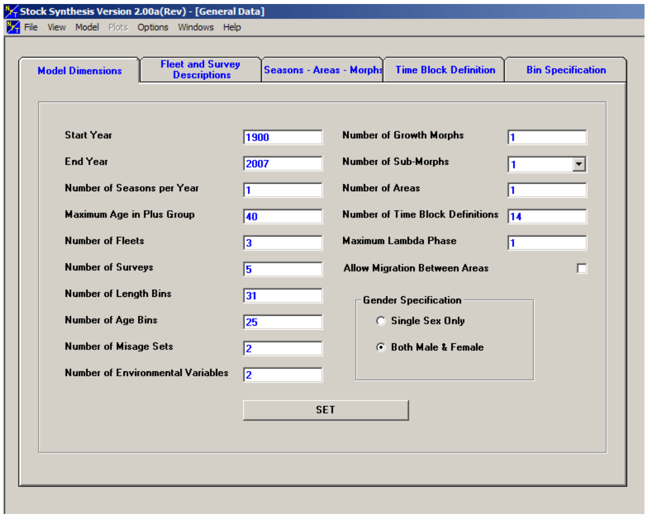
The SS graphical interface uses the same general approach as other stock assessment models in the NOAA Fisheries Assessment Toolbox. The GUI reads and writes DOS text files that are identical in content to the text files described in this document. When the GUI is started, the user sees the Input Files screen. Click on File-Open Existing SS File, navigate to the folder containing the target files, and open STARTER.SS. The GUI opens all necessary files.



After reading the files, the user can then select one of the 5 windows for reviewing and editing input information. These windows are:

1. Starter file
2. General data (basically model dimensions)
3. Forecast specifications
4. Observed data (most information from the dat file)
5. Control parameters (most information from the ctl file).

Each window contains a multiple document interface with different tabs for different categories of information.



The GUI contains an internal editing window that allows for editing a cell, or continuously editing within a row or column. Marking a block of cells or clicking on a row or column header calls up a cut – paste option. In addition, blocks of information can be cut from external text editors or spreadsheets and pasted into the GUI cells, but only if the size of the block conforms. Also the user can resize the width of columns to allow viewing the most relevant columns more easily.

If you give the GUI a command that results in resizing arrays (such as changing the selectivity pattern to be used for a particular fishery so that the required number of parameters is changed) then the set of new parameters will be set to blank. The user could then select a block of control specifications from SS3-examples.xls for the newly invoked selectivity function and paste them into this blank area.

When the user is ready to run the program, select Model – Run. The GUI will then save all the input files, run the program, return to the GUI, scan the output files, and return control to the user for viewing the output.

Before using the GUI, it is wise to save a back-up copy of all input files. This is because all custom comments you have placed in these text files will be lost when they are read and rewritten by the GUI. Also, possible GUI crashes may cause loss of some information when the input files are being written.

Future developments with the GUI may include more internal business rules to check for illegal or illogical combinations of input choices, more context-sensitive help, scanning of more output files, etc.

11.4 Debugging Tips

When SS input files are causing the program to crash or fail to produce sensible results, there are a few steps that can be taken to diagnose the problem. Before trying the steps below, examine the ECHOINPUT.SSO file. It is highly annotated, so you should be able to see if SS is interpreting your input files as you intended.

1. Set the turn_off_phase switch to 0 in the STARTER.SS file. This will cause the mode to not attempt to adjust any parameters and simply converges a dummy parameter. It will still produce a REPORT.SSO file, which can be examined to see what has been calculated from the initial parameter values.

2. Turn the verbosity level to 2 in the STARTER.SS file. This will cause the program to display the value of each likelihood component to the screen on each interation. So it the program is creating an illegal computation (e.g. divide by zero), it may show you which likelihood component contains the problematic calculation. If the program is producing a REPORT.SSO file, you may then see which observation is causing the illegal calculation.
3. Run the program with the command SS3 >>SSpipe.txt. This will cause all screen display to go to the specified text file (note, delete this file before running because it will be appended to). Examination of this file will show detailed statements produced during the reading and preprocessing of input files.
4. *CHECKUP.SSO*: This file can be written during the first iteration of the program. It contains details of selectivity and other calculations as an aid to debugging model problems.
5. If SS fails to achieve a proper Hessian it exits without writing the detailed outputs in the FINAL_SECTION. If this happens, you can do a run with the –nohess option so you can view the report.sso to attempt to diagnose the problem.

11.5 Keyboard Tips

Typing “N” during a run will cause ADMB to immediately advance to the next phase of estimation.

Typing “Q” during a run will cause ADMB to immediately go to the final phase. This bypasses estimation of the Hessian and will produce all of the SS outputs, which are coded in the FINAL_SECTION.

11.6 Running MCMC

Run SS3

- This gives MPD estimates, report file, Hessian matrix and the .cor file
- (Recommended) Look for parameters stuck on bounds which will degrade efficiency of MCMC implementation
- (Recommended) Look for very high correlations that may degrade the efficiency of MCMC implementation

Run SS3 with arguments -mcmc xxxx -mcsave yyyy

- Where: xxxx is the number of iterations for the chain, and yyyy is the thinning interval (1000 is a good place to start)
- MCMC chain starts at the MPD values
- (Recommended) Remove existing .psv files in run directory to generate a new chain.
- (Recommended) Set DOS run detail switch in starter file to 0; reporting to screen will dramatically slow MCMC progress
- (Optional) Add -nohess to use the existing Hessian file without re-estimating
- (Optional) Add -noest -nohess -mcpin myfile.xxx starts the MCMC chain from any values you want (file name "myfile.xxx" matching exactly the .par file format)

- (Optional) Add -noest -nohess and modify starter file so that run will now start from the converged (or modified) parameter estimates in "ss3.par"

Run SS3 with argument -mceval

- This generates the posterior output files.
- (Optional) Modify starter file entries to add a burn-in and thinning interval above and beyond the ADMB thinning interval applied at run time.
- (Recommended) MCMC always begins with the MPD values and so a burn-in >0 should always be used.
- This step can be repeated for alternate forecast options (e.g. catch levels) without repeating step 2.

(Optional) Run SS3 with arguments -mcr -mcmc xxxx -mcsave yyyy ...

- This restarts and extends an uninterrupted chain previously completed (note that any intermediate runs without the -mcr command in the same directory will break this option).

NOTES:

When SS switches to MCMC or MCEVAL mode, it sets all the bias adjustment factors to 1.0 for any years with recruitment deviations defined. SS does not create a report file after completing MCMC because it would show values based on the last MCMC step.

11.7 Using R to view model output

A collection of functions developed as a package, r4ss, for the statistical software R has been created to explore SS model output. The functions include tools for summarizing and plotting results, manipulating files, visualizing model parameterizations, and various other tasks. Currently, information on the code can be found at code.goole.com/p/r4ss which facilities exploration of the functions, code, information on any major changes, and allows for submission of questions or issues. Specific information on the R package, r4ss, can be found on the CRAN website (cran.r-project.org/web/packages/r4ss/index.html). The software package is under constant development to maintain compatibility with new versions of SS and to improve functionality. The code in package form can be installed within R using the commands:

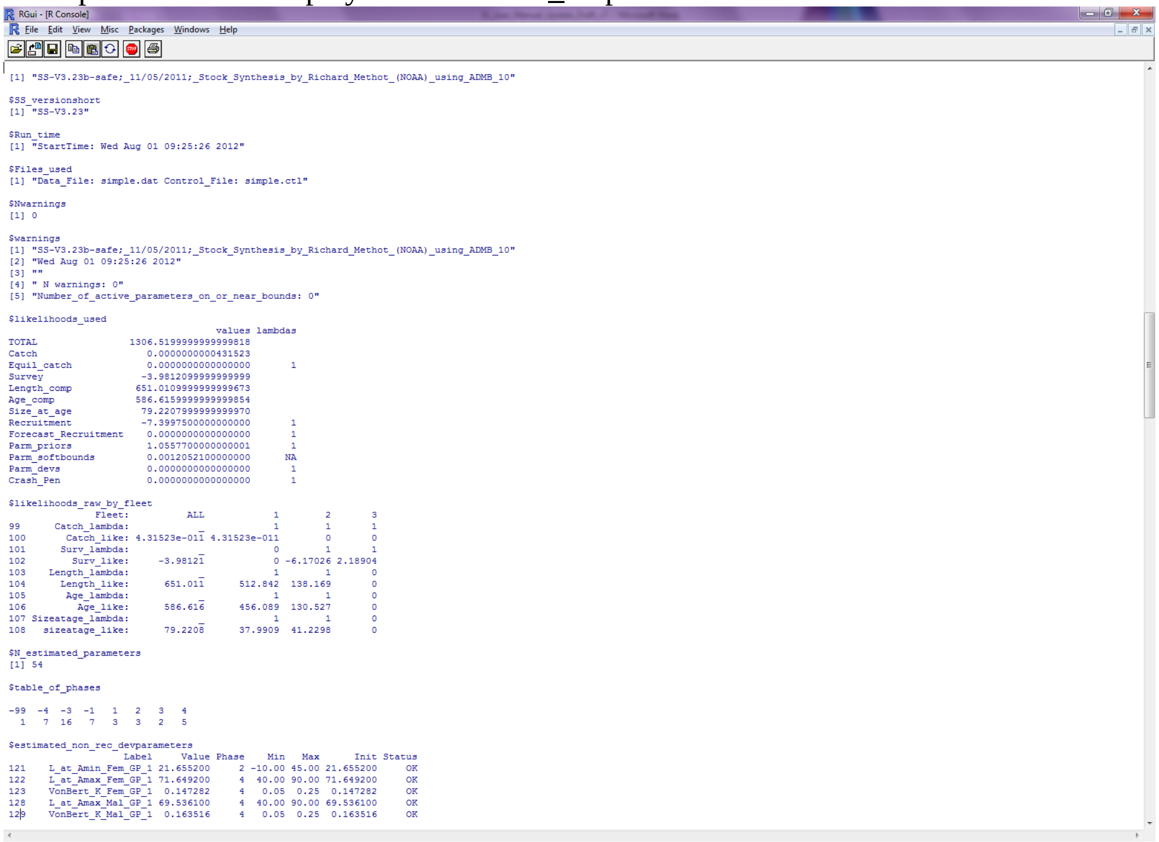
```
> install.packages("r4ss")
> library(r4ss).
```

After the package has been install updates can be obtained by:

```
> update_r4ss_files()
```

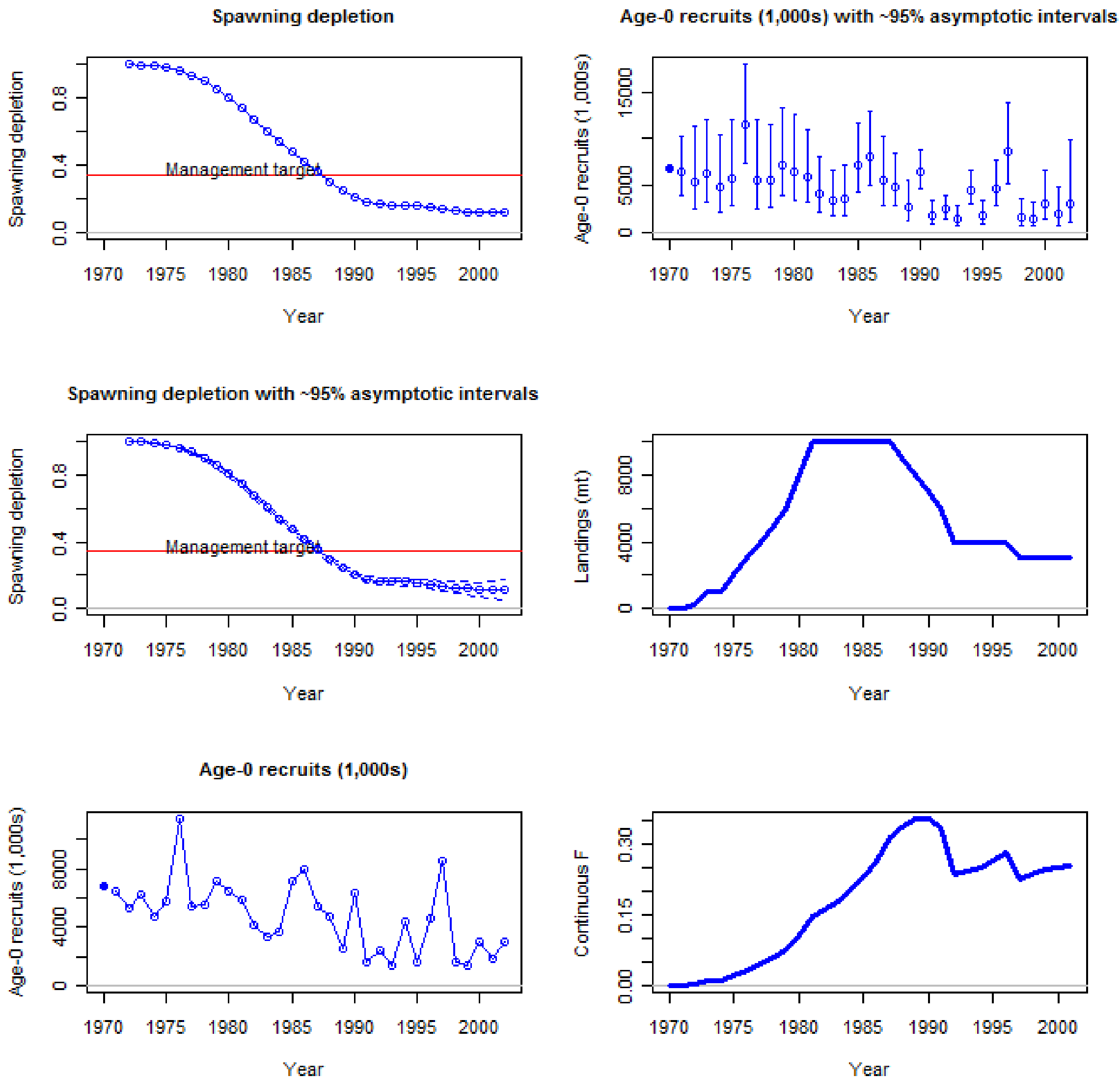
Two of the most commonly useful functions for model diagnostics are SS_output and SS_plots. After running a model using SS the report can be read into R by the SS_output function which stores quantities in a list with named objects. The SS_plots function creates a series of plots that are useful to visual the model fit to the data and the estimated and fixed parameters.

Example of the data displayed used the SS_output function.




```
[1] "SS-V3.23b-safe;_11/05/2011;_Stock_Synthesis_by_Richard_Methot_(NOAA)_using_ADMB_10"

$SS_versionshort
[1] "SS-V3.23"

$Run_time
[1] "StartTime: Wed Aug 01 09:25:26 2012"

$Files_used
[1] "Data_File: simple.dat Control_File: simple.ctl"

$Nwarnings
[1] 0

$warnings
[1] "SS-V3.23b-safe;_11/05/2011;_Stock_Synthesis_by_Richard_Methot_(NOAA)_using_ADMB_10"
[2] "Wed Aug 01 09:25:26 2012"
[3] ""
[4] " N warnings: 0"
[5] "Number_of_active_parameters_on_or_near_bounds: 0"

$likelihoods_used
                               values lambdas
TOTAL                1306.5199999999999818
Catch                   0.0000000000431523
Equil_catch             0.0000000000000000       1
Survey                 -3.9812099999999999
Length_comp           651.0109999999999673
Age_comp              586.6159999999999854
Size_at_age            79.2207999999999970
Recruitment            -7.3997500000000000       1
Forecast_Recruitment    0.0000000000000000       1
Parm_priors             1.0557700000000001       1
Parm_softbounds         0.0012052100000000      NA
Parm_devs               0.0000000000000000       1
Crash_Pen               0.0000000000000000       1

$likelihoods_raw_by_fleet
               Fleet:          ALL            1        2       3
99      Catch_lambda:            _            1        1       1
100       Catch_like: 4.31523e-011 4.31523e-011        0       0
101      Surv_lambda:            _            0        1       1
102        Surv_like:     -3.98121            0 -6.17026 2.18904
103    Length_lambda:            _            1        1       0
104      Length_like:      651.011      512.842  138.169       0
105       Age_lambda:            _            1        1       0
106         Age_like:      586.616      456.089  130.527       0
107 Sizeatage_lambda:            _            1        1       0
108   sizeatage_like:      79.2208      37.9909  41.2298       0

$N_estimated_parameters
[1] 54

$table_of_phases

-99  -4  -3  -1   1   2   3   4
  1   7  16   7   3   3   2   5

$estimated_non_rec_devparameters
                    Label     Value Phase    Min   Max      Init Status
121    L_at_Amin_Fem_GP_1 21.655200     2 -10.00 45.00 21.655200     OK
122    L_at_Amax_Fem_GP_1 71.649200     4  40.00 90.00 71.649200     OK
123    VonBert_K_Fem_GP_1  0.147282     4   0.05  0.25  0.147282     OK
128    L_at_Amax_Mal_GP_1 69.536100     4  40.00 90.00 69.536100     OK
129    VonBert_K_Mal_GP_1  0.163516     4   0.05  0.25  0.163516     OK
```

Example of plots created using the SS_plots funtions.



The functions included in r4ss ranging from general use to functions developed for specific model applications:

Core functions:	
SS_output	A function to create a list object for the output from Stock Synthesis
SS_plots	Plot many quantities related to output from Stock Synthesis
Function for updating files between package revisions	
update_r4ss_files	Updates r4ss files to newest versions on web.
Plot functions called by SS_plots:	
SSplotBiology	Plot biology related quantities.

SSplotCatch	Plot catch related quantities.
SSplotCohorts	Plot cumulative catch by cohort.
SSplotComps	Plot composition data and fits.
SSplotData	Timeline of presence/absence of data by type, year, and fleet.
SSplotDiscard	Plot fit to discard fraction.
SSplotIndices	Plot indices of abundance and associated quantities.
SSplotMnwt	Plot mean weight data and fits.
SSplotMovementMap	Show movement rates on a map.
SSplotMovementRates	Show movement rates on a map.
SSplotNumbers	Plot numbers-at-age related data and fits.
SSplotRecdevs	Plot recruitment deviations
SSplotRecdist	Plot of recruitment distribution among areas and seasons
SSplotSelex	Plot selectivity
SSplotSpawnrecruit	Plot spawner-recruit curve.
SSplotSPR	Plot SPR quantities.
SSplotTags	Plot tagging data and fits
SSplotTimeseries	Plot timeseries data
SSplotYield	Plot yield and surplus production.
SS_html	Create HTML files to view figures in browser.
SS_fitbiasramp	Estimate bias adjustment for recruitment deviates
Model Comparisons:	
SSgetoutput	Get output from multiple Stock Synthesis models.
SSsummarize	Summarize the output from multiple Stock Synthesis models.
SSplotComparisons	plot model comparisons
SStableComparisons	make table comparing quantities across models
addSSsummarize	Add a model to the list of models to compare
Additional tools for plotting model output	
SSplotpars	Plot distributions of priors, posteriors, and estimates.
SSplotProfile	Plot likelihood profile results
Functions related to MCMC diagnostics	
mcmc.nuisance	Summarize nuisance MCMC output
mcmc.out	Summarize, analyze and plot key MCMC output.
SSgetMCMC	Read MCMC output.
SSplotMCMC_ExtraSelex	Plot uncertainty around chosen selectivity ogive from MCMC.
Interactive tools for exploring functional forms:	
movepars	Explore movement parameterizations
selfit	A function to visual parameterization of double normal and double logistic selectivity in Stock Synthesis

selfit_spline	Sisualize parameterization of cubic spline selectivity in SS
sel.line	A function for drawing selecitivity curves
File manipulation for inputs:	
SS_readctl	Read control file (incomplete)
SS_readdat	Read data file
SS_readforecast	Read forecast file
SS_readstarter	Read starter file
SS_writectl	Write control file (incomplete)
SS_writedat	Write data file
SS_writeforecast	Write forecast file
SS_writestarter	Write starter file
SS_makedatlist	Make a list for SS data
SS_parlines	Get parameter lines from Stock Synthesis control file
SS_changepars	Change parameters in the control file.
SSmakeMmatrix	Create inputs for entering a matrix of natural mortality by age and year
SS_profile	Run a likelihood profile in Stock Synthesis (incomplete)
File manipulation for outputs:	
SS_recdevs	Insert a vector of recruitment deviations into the control file.
SS_splitdat	Split apart bootstrap data to make input file.
SSchangeYears	The function is currently incomplete
Minor plotting functions (some borrowed from other folks)	
bubble3	Create a bubble plot.
make_multifig	Create multi-figure plots.
plotCI	Plot points with confidence intervals.
rich.colors.short	Make a vector of colors.
stackpoly	Function "stackpoly" by Jim Lemon from "plotrix" package
mountains	Make shaded polygons with a mountain-like appearance
Really specialized functions:	
DoProjectPlots	Make plots from Rebuilder program
IOTCmove	Make a map of movement for a 5-area Indian Ocean model
SSFishGraph	A function for converting Stock Synthesis output to the format used by FishGraph
TSCplot	Create a plot for the TSC report

12. Special Set-ups

12.1 Continuous seasonal recruitment

It is awkward in SS to set up a seasonal model such that recruitment can occur with similar and independent probability in any season of any year. Consequently, some users have attempted to setup SS so that each quarter appears as a year. They have set up all the data and parameters to treat quarters as if they were years (i.e. each still has a duration of 1.0 time step). This can work, but requires that all rate parameters be re-scaled to be correct for the quarters being treated as years.

Another option is available. If there is one season per year and the season duration is set to 3 (rather than the normal 12), then the season duration is calculated to be 3/12 or 0.25. This means that the rate parameters can stay in their normal per year scaling and this shorter season duration makes the necessary adjustments internally. Some other adjustments to make when doing quarters as years include:

- re-index all "year seas" inputs to be in terms of quarter-year because all are now season 1; increase endyr value in sync with this
- increase max age because age is now in quarters
- in the age error definitions, increase the number of entries in accord with new max age
- in the age error definitions, recode so that each quarter-age gets assigned to the correct agebin; This is because the age data are still in terms of agebins; i.e. the first 4 entries for quarter-ages 1 through 4 will all be assigned to agebin 1.5; the next four to agebin 2.5; you cannot accomplish the same result by editing the age bin values because the stddev of ageing error is in terms of agebin
- in the control file, multiple the natM age breakpoints and growth AFIX values by 1/seasdur
- decrease the R0 parameter starting value because it is now the average number of recruitments per qtryear
- edit the rec_dev start and endyrs to be in terms of qtryear
- edit any age selectivity parameters that refer to age to now refer to qtrage
- if there needs to be some degree of seasonality to recruitment or some parameter, then you could create a cyclic pattern in the environmental input and make recruitment or some other parameter a function of this cyclic pattern

A good test showing comparability of the 3 approaches to setting up a quarterly model should be done.

13. Change Log

This section has been removed from the user manual. Information on changes to SS is now recorded in the spreadsheet database, SS_Changes.xlsx. Fields include date, version number, category (e.g. growth, selectivity), type (e.g. new, clarify, fix). Occasional model tips will be added with the type="Tip".

14. Appendix A: Recruitment Variability and Bias Corrections

Recruitments in SS are defined as lognormal deviates around a log-bias adjusted spawner-recruitment curve. The magnitude of the log-bias adjustment is calculated from the level of σ_R, which is the standard deviation of the recruitment deviations (in log-space). There are 5 segments of the time series in which to consider the effect of the log-bias adjustment: virgin; initial equilibrium; early data-poor period; data-rich period; very-recent/forecast. The choice of break points between these segments need not correspond directly with the settings for the bias adjustment, although some alignment might be desired. Methot and Taylor (2011) provide more detailed discussion of the bias adjustment than what is provided below but do not address the separation of time periods into separate segments. The approach is illustrated with figures associated with a recent assessment for darkblotched rockfish (Gertseva and Thorson, 2013).

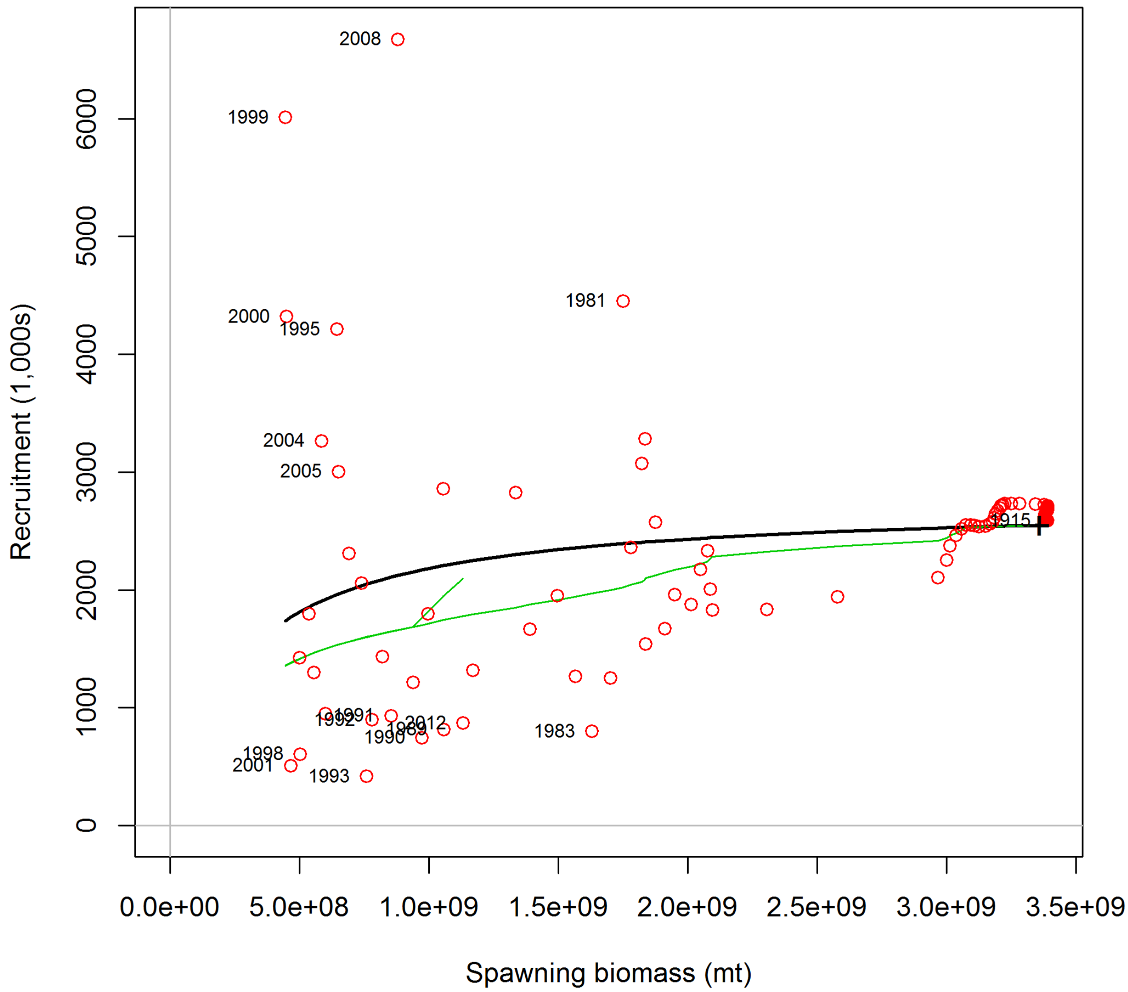


Figure A.1. Spawner-recruitment relationship for darkblotched rockfish (Gertseva and Thorson, 2013). Red points represent estimated recruitments, the solid black line is the stock-recruit relationship and the green line represents the adjustment to this relationship after adjustment to account for the lognormal distribution associated with each year. The "+" symbol labeled 1915 near the right side represents both the virgin and initial equilibrium of the model. The numerous red points close to the initial conditions correspond to the early years of the model with low harvest rates.

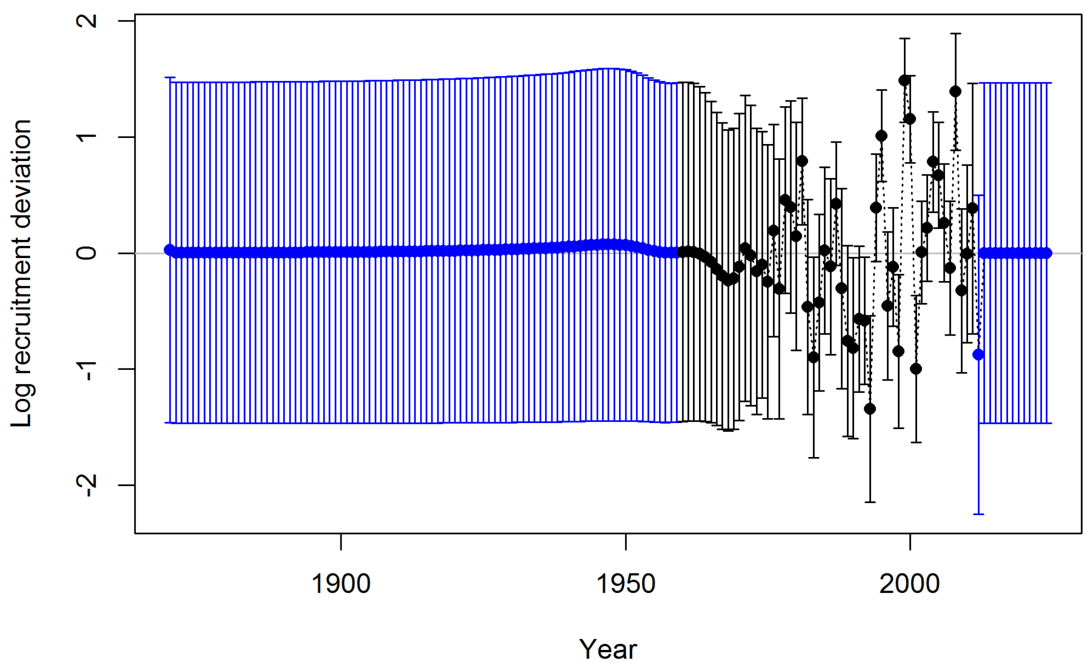


Figure A.2. Timeseries of log recruitment deviations for darkblotched rockfish with 95% uncertainty intervals. The start year of the model is 1915, but recruitment deviations are estimates starting in 1870. The 45 deviation estimates for 1870–1914 inform the age structure used in the start year. The black color for the years 1960–2011 indicates the "main" recruitment deviation vector, while the blue color for the years 1870–1959 and 2012–2024 indicates the "early" and "late/forecast" recruitment deviation vectors, respectively.

1. **Virgin** – the R_0 level of recruitment is a parameter of the spawner-recruitment curve. This recruitment and the corresponding spawning biomass, S_0, are expected to represent the long-term arithmetic mean.
2. **Initial equilibrium** – the level of recruitment is typically maintained at the R_0 level even though the initial equilibrium catch will reduce the spawning biomass below the virgin level. If steepness is moderately low or the initial F is high, then the lack of response in recruitment level may appear paradoxical. The logic is that building in the spawner-recruitment response to initial F would significantly complicate the calculations and would imply that the initial equilibrium catch level had been going on for multiple generations. If the lack of response is considered to be problematic in a particular application, then start the model at an earlier year and with a lower initial equilibrium catch so that the dynamics of the spawner-recruitment response get captured in the early period, rather than getting lost in the initial equilibrium.
3. **Early data-poor period** – This is the early part of the time series where the only data typically are landed catch. There are no data to inform the model about the specific year-to-year fluctuations in recruitment, although the ending years of this period will begin to be influenced by the data. The "early time period" is not a formal concept.

It is up to the user to decide whether to start estimating recruitment deviations beginning with the first year of the model, or to delay such estimation until the data become more informative. Modeling recruitment deviations in this period may lead to a more realistic portrayal of the uncertainty in depletion, but can also lead to spurious patterns in estimated recruitments that may be driven by the fit to index data or other sources that wouldn't be expected to have accurate information on recruitment.

Option A: do not estimate recruitment deviations during this early period. During years prior to the first year of recruitment deviations, the model will set the recruitment equal to the level of the spawner-recruitment curve. Thus, it is a mean-based level of recruitment. Because these annual parameters are fixed to the level of the spawner-recruitment curve, they have no additional uncertainty and make no contribution to the variance of the model.

This approach may produce relatively large, or small, magnitude deviations at the very beginning of the subsequent period, as the model "catches up" to any slight signal that could not be captured through estimated deviations in the early data-poor period. There may be some effect on the estimate of R_0 as a result of lack of model flexibility in balancing early period removals with signal in the early portion of the data-rich period.

Option B: estimate recruitment deviations for all the early years. Each of these recruitment deviations is now a dev parameter so will have a variance that contributes to the total model variance. The estimated standard deviation of each of these dev parameters should be similar to σ_R because σ_R is the only constraint on these parameters (however, the last few in the sequence will begin to feel the effect of the data so may have lower standard deviations).

4. **Data-rich period**: Here the data inform the model on the year-to-year level of recruitment. These fluctuations in recruitment are assumed to have a lognormal distribution around the log-bias adjusted spawner-recruitment curve. The level of σ_R input to the model should match this level of fluctuation to a reasonable degree. Because the recruitments are lognormal, they produce a mean biomass level that is comparable to the virgin biomass and thus the depletion level can be calculated without bias. However, if the early period has recruitment deviations estimated by MPD, then the depletion levels during the early part of the data-rich period may have some lingering effect of negative bias during the early time period.

   The level of σ_R should be at least as large as the level of variability in these estimated recruitments. If too high a level of σ_R is used, then a bias can occur in the estimate of spawner-recruitment steepness, which determines the trend in recruitment. This occurs when the early recruitments are taken directly from the spawner-recruitment curve, so are mean unbiased, then the later recruitments are estimated as deviations from the log-bias adjusted curve. If σ_R is too large, then the bias-adjustment is too large, and the model may compensate by increasing steepness to keep the mean level of recent recruitments at the correct level.

5. **Recent Years/Forecast**: Here the situation is very similar to the early time period in that there are no data to inform the model about the year-to-year pattern in recruitment fluctuations so all devs will be pulled to a zero level in the MPD. The structure of SS creates no sharp dividing line between the estimation period and the forecast period. In many cases one or more recruitments at the end of the time series will lack appreciable signal in the data and should therefore be treated as forecast recruit deviations. To the degree that some variability is observed in these recruitments, partial or full bias correction may be desirable for these devs separate from the purely forecast devs, there is therefore an additional control for the level of bias correction applied to forecast deviations occurring prior to endyear+1.

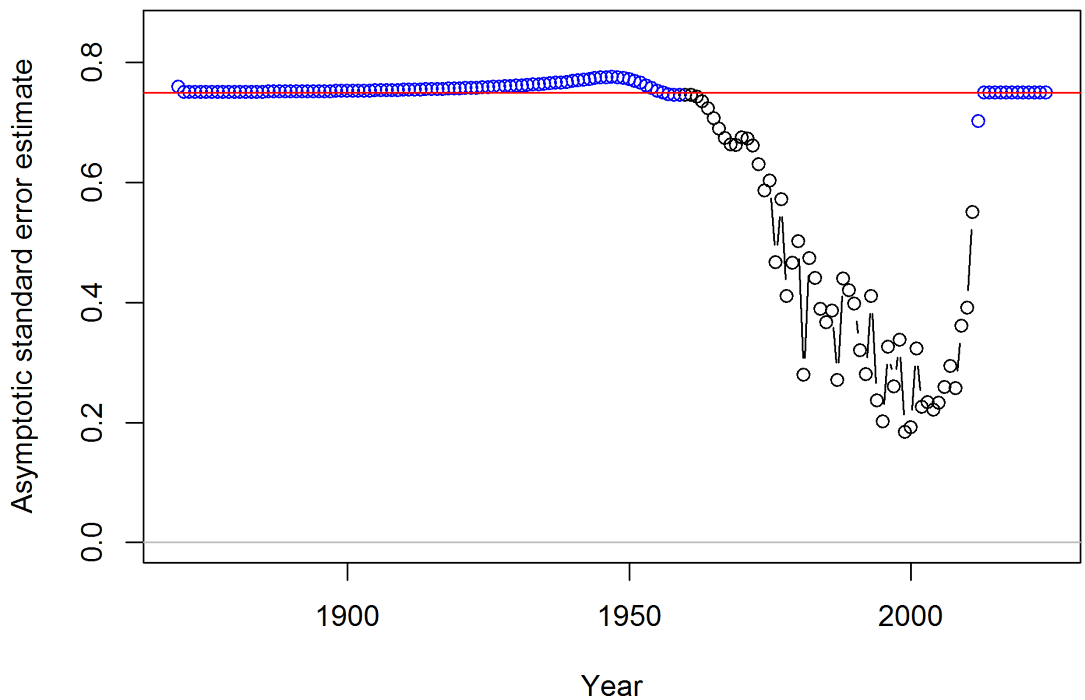


Figure A.3. Timeseries of standard error estimates for the log recruitment deviations for darkblotched rockfish with 95% uncertainty intervals. As in Figure A.2, the black color indicates the main recruitment period. This period with lower standard error is associated with higher variability among deviations (Figure A.2). The red line at 0.75 indicates the σ_R value in this model.

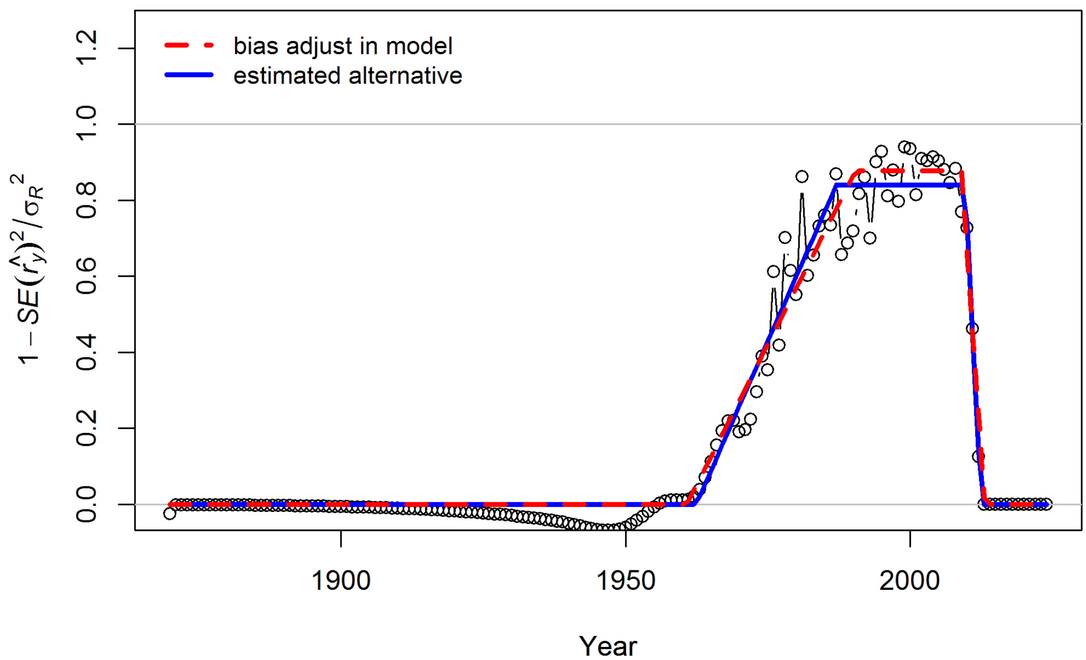


Figure A.4. Transformation of the standard error estimates (shown in Figure A.3) for darkblotched rockfish following the approach suggested by Methot and Taylor (2011). These values were used to set the 5 values controlling the degree of bias adjustment (as a fraction of $\sigma_R^2/2$) to account for differences in the mean and median of the lognormal distribution from which the recruitment deviations are drawn. The red line indicates a bias adjustment of 0 up to the 1960.75, ramping up to a maximum adjustment level of 0.877 for the period 1990.4–2008.98, and reducing back to 0 starting in 2013.08. Note that these values controlling the bias adjustment need not be integer year values. Also the break points in the bias adjustment function need not match the break points between early, main, and late/forecast recruitment deviation vectors (indicated by blue and black colors in Figures A.2 and A.3). The blue line indicates a functional form that minimizes the sum of squared differences between the bias adjustment function and the transformed standard error values. The subtle differences between red and blue lines are unlikely to have any appreciable effect on the model results.

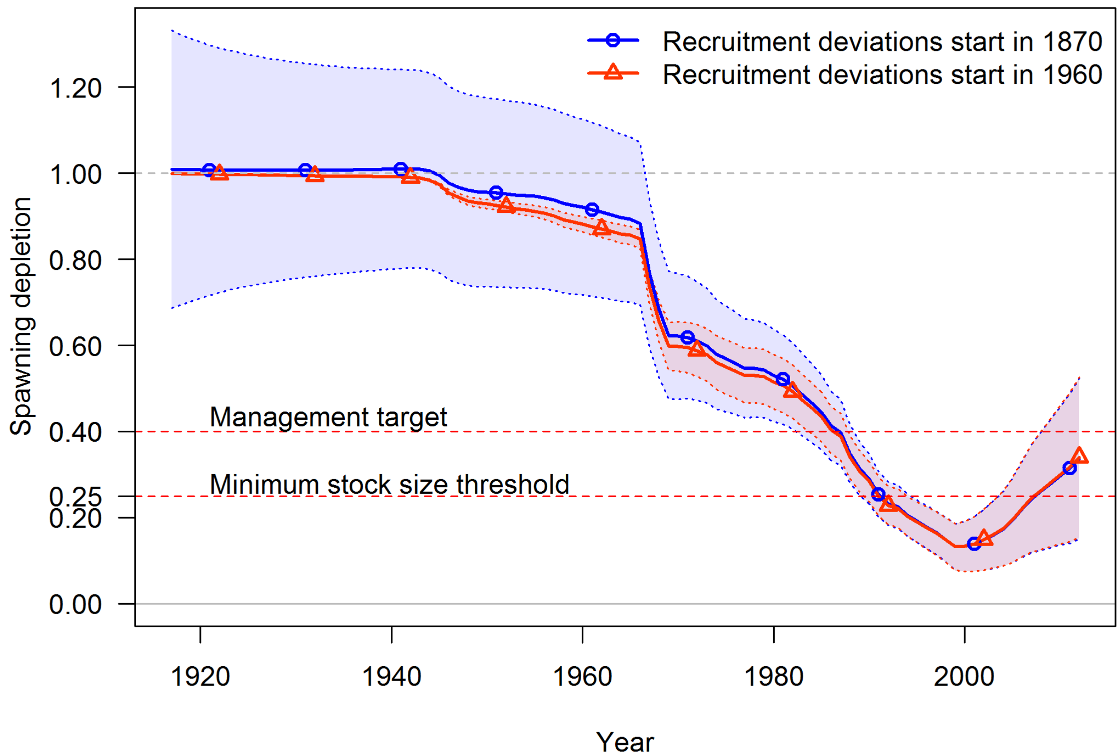


Figure A.5. Comparison of timeseries of spawning depletion for darkblotched rockfish models with early recruitment deviations (starting in 1870) and without early deviations (only main recruitment deviations starting in 1960). The point estimates are similar, but the 95% uncertainty intervals are substantially different. With no recruitment deviations for the early period, the estimates of spawning depletion in the early years are very precise and uncertainty increases as the stock moves into the data rich period. In contrast, the addition of the early recruitment deviations results in a large uncertainty in spawning depletion for the early years and an increase in precision as the stock moves into the data rich period. In this application, the uncertainty associated with the recent years is independent of the assumptions about early recruitments.

Issues with Including Environmental Effects:

The expected level of recruitment is a function of spawning biomass, an environmental time series, and a log-bias adjustment.

$$E(\text{Recruitment}) = f(\text{SpBio}) * \exp(\beta*\text{envdata}) * \exp(-0.5*\sigma_R^2)$$

σ_R is the variability of the deviations, so it is in addition to the variance "created" by the environmental effect. So, as more of the total recruitment variability is explained by the environmental effect, the residual σ_R should be decreased. The model does not do this automatically.

The environmental effect is inherently lognormal. So when an environmental effect is included in the model, the arithmetic mean recruitment level will be increased above the level predicted by f(SpBio) alone. The consequences of this have not yet been thoroughly investigated, but there probably should be another bias correction based on the variability of the environmental data as scaled by the estimated linkage parameter, β.

It is also problematic that the environmental effect time series used as input is assumed to be measured without error.

The preferred approach to including environmental effects on recruitment is not to use the environmental effect in the direct calculation of the expected level of recruitment. Instead, the environmental data would be used as if it was a survey observation of the recruitment deviation. This approach is similar to using the environmental index as if it was a survey of age 0 recruitment abundance because by focusing on the fit to the deviations it removes the effect of SpBio on recruitment. In this alternative, the σ_R would not be changed by the environmental data; instead the environmental data would be used to explain some of the total variability represented by σ_R. This approach may also allow greater uncertainty in forecasts, as the variability in projected recruitments would reflect both the uncertainty in the environmental observations themselves and the model fit to these observations.

Initial Age Composition – If the first year with recruitment deviations is set less than the start year of the model, then these early deviations will modify the initial age composition. The amount of information on historical recruitment variability certainly will degrade as the model attempts to estimate deviations for older age groups in the initial equilibrium. So the degree of bias correction is reduced linearly in proportion to age so that the correction disappears when maximum age is reached. The initial age composition approach normally produces a result that is indistinguishable from a configuration that starts earlier in the time series and estimates a longer time series of recruitments. However, because the initial equilibrium is calculated from a recruitment level unaffected by spawner-recruitment steepness and initial age composition adjustments are applied after the initial equilibrium is calculated, it is possible that the initial age composition approach will produce a slightly different result than if the time series was started earlier and the deviations were being applied to the recruitment levels predicted from the spawner-recruitment curve.

15. Appendix B: Forecast Module

Benchmark and Forecast Module in Stock Synthesis
Dec 14, 2010

14.1 Introduction

Version 3.20 of Stock Synthesis (SS) introduced substantial upgrades to the benchmark and forecast module. The general intent was to make the forecast outputs more consistent with the requirement to set catch limits that have a known probability of exceeding the overfishing limit. In addition, this upgrade addressed several inadequacies with the previous module, including:

- The average selectivity and relative F was the same for the benchmark and the forecast calculations;

- The biology-at-age in endyr+1 was used as the biology for the benchmark, but biology –at-age propagated forward in the forecast if there was time-varying growth;
- The forecast module had a kluge approach to calculation of OFL conditioned on previously catching ABC;
- The forecast module implementation of catch caps was incomplete and applied some caps on a seasonally, rather than the more logical annual basis;
- The Fmult scalar for fishing intensity presented a confusing concept for many users;
- No provision for specification of catch allocation among fleets ;
- The forecast allowed for a blend of fixed input catches and catches calculated from target F; this is not optimal for calculation of the variance of F conditioned on a catch policy that sets ACLs.

The V3.20 module addressed these issues by:

- Providing for unique specification of a range of years from which to calculate average selectivity for benchmark, average selectivity for forecast, relative F for benchmark, and relative F for forecast;
- Create a new specification for the range of years over which to average size-at-age and fecundity-at-age for the benchmark calculation. In a setup with time-varying growth, it may make sense to do this over the entire range of years in the time series. Note that some additional quantities still use their endyr values, notably the migration rates and the allocation of recruitments among areas. This will be addressed shortly;
- Create a multiple pass approach that rectifies the OFL calculation;
- Improve the specification of catch caps and implement specification of catch allocations so that there can be an annual cap for each fleet, an annual cap for each area, and an annual allocation among groups of fleets (e.g. all recreational fleets vs. all commercial fleets);
- Introduce capability to have implementation error in the forecast catch (single value applied to all fleets in all seasons of the year).

14.2 Multiple Pass Forecast

The most complicated aspect of the changes is with regard to the multiple pass aspect of the forecast. This multiple pass approach is needed to calculate both OFL and ABC in a single model run. More importantly, the multiple passes are needed in order to mimic the actual sequence of assessment-management action – catch over a multi-year period. The first pass calculates OFL based on catching OFL each year, so presents the absolute maximum upper limit to catches. The second pass forecasts a catch based on a harvest policy, then applies catch caps and allocations, then updates the F's to match these catches. In the third pass, stochastic recruitment and catch implementation error are implemented and SS calculates the F that would be needed in order to catch the adjusted catch amount previously calculated in the second pass. With this approach, SS is able to produce improved estimates of the probability that F would exceed the overfishing F. In effect it is the complement of the P* approach. Rather than the P* approach that calculates the stream of annual catches that would have an annual probability of F>Flimit, SS calculates the expected time series of P* that would result from a specified harvest policy implemented as a buffer between Ftarget and Flimit.

The sequence of multiple forecast passes is as follows:

1. Pass 1 (aka Fcast_Loop1)
    a. Loop Years
        i. SubLoop (aka ABC_Loop) = 1
            1. R=f(SSB) with no deviations
            2. F=Flimit
            3. Fixed input catch amounts ignored
            4. No catch adjustments (caps and allocations)
            5. No implementation error
            6. Result: OFL conditioned on catching OFL each year
2. Pass 2
    a. Loop Years
        i. SubLoop = 1
            1. R=f(SSB) with no deviations
            2. F=Flimit
            3. Fixed input catch amounts ignored
            4. No catch adjustments (caps and allocations)
            5. No implementation error
            6. Result: OFL conditioned on catching ABC previous year. Stored in std_vector
        ii. SubLoop = 2
            1. R=f(SSB) with no deviations
            2. F=Ftarget to get target catch for each fleet in each season
            3. Fixed input catch amounts replace catch from step 2
            4. Catch adjustments (caps and allocations) applied on annual basis (after looping through seasons and areas within this year). These adjustments utilize the logistic joiner approach common in SS so the overall results remain completely differentiable
            5. No implementation error
            6. Result: ABC as adjusted for caps and allocations
        iii. SubLoop = 3
            1. R=f(SSB) with no deviations
            2. F=adjusted to match adjusted catches from Subloop=2
            3. No implementation error
            4. Result: Adjusted survivors to pass to the next year

3. Pass 3
    a. Loop Years
        i. Subloop = 3
            1. R=F(SSB) with recruitment deviations
            2. Catches from Pass 2 multiplied by the random term for implementation error
            3. F=adjusted to match the catch*error while taking into account the random recruitments. This is most easily visualized in a MCMC context where the recruitment deviation and the implementation error deviations take on non-zero values in each instance. In MLE, because the forecast recruitments and implementation error are estimated parameters with variance, their variance still propagates to the derived quantities in the forecast.
            4. Result: Values for F, SSB, Recruitment, Catch are stored in std-vectors
                a. In addition, the ratios F/Flimit and SSB/SSBlimit or SSB/SSBtarget are also stored in std_vectors.
                b. Estimated variance in these ratios allows calculation of annual probability that F>Flimit or B<Blimit. This is essentially the realized P* conditioned on the specified harvest policy.

14.3 Example Effect on Correlations

An example that illustrates the above process was conducted. The situation was a low M, late-maturing species, so changes are not dramatic. The example conducted a 10 year forecast and examined correlations with derived quantities in the last year of the forecast. This was done once with the full set of 3 passes as described above, and again with only 2 passes and stochastic recruitment occurring in pass 2, rather than 3. This alternative setup is more similar to forecasts done using previous model versions.

	3 Forecast Passes with F from ABC and random recruitment				2 Forecast Passes with Catch from Target F and equil recruitment		
	Factor X	*Factor Y*	*corr*		*Factor X*	*Factor Y*	*corr*
A1	F_2011	RecrDev_2002	-0.126	A2	F_2011	RecrDev_2002	0.090
B1	F_2011	Recr_2002	0.312	B2	F_2011	Recr_2002	0.518
C1	ForeCatch_2011	RecrDev_2002	0.000	C2	ForeCatch_2011	RecrDev_2002	0.129
D1	ForeCatch_2011	Recr_2002	0.455	D2	ForeCatch_2011	Recr_2002	0.555

Correlation A2 shows a small positive correlation between the recruitment deviation in 2002 and the F in 2011. This is probably due to the fact that a positive deviation in recruitment in 2002 will reduce the chances that the biomass in 2011 will be below the inflection point in the control rule. This occurs because in calculating catch from F, the model effectively "knows" the future recruitments. I predict that this B1 correlation would be near zero if there was no inflection in the control rule.

Correlation A1 shows this turning into a negative correlation. This is because the future catches are first calculated from equilibrium recruitment, then when random recruitments are implemented, a positive recruitment deviation will cause a negative deviation in the F needed to catch that now "fixed" amount of future catch.

Correlations B1 and B2 are in terms of absolute recruitment, not recruitment deviation. Now overall model conditions that cause a higher absolute recruitment level will also result in a higher forecast level. No surprise there, and the correlation is stronger when variance is based on catch is calculated from F (B2).

Correlation C2 shows a positive correlation between recruitment deviation in 2002 and forecast catch in 2011. However, correlation C1 is 0.0 because the forecast catch in 2011 is set based on equilibrium recruitment and is not influenced by the recruitment deviations.

14.4 Future Work

- More testing with high M, rapid turnover conditions
- Testing without inflection in control rule
- Consider separating implementation error into a pass #4 so results will more clearly show effect of assessment uncertainty separate from implementation uncertainty
- Consider adding a random "assessment" error which essentially is a random variable that scales population abundance before passing into the forecast stage. Complication is figuring out how to link it to the correlated error in the benchmark quantities
- Because all of these calculations occur only in the sdphase or the mceval phase, it would be feasible for mceval calls to add an additional pass that is implemented many times and in which random forecast recruitment draws are made.
- Factors like selectivity and fleet relative F levels are calculated as an average of these values during the time series. This is internally consistent if these factors do not vary during the time series (although clearly this is a stiff model that will underestimate process variance). However, if these factors do vary over time, then the average used for the forecast will under-represent the variance. A better approach would be to set up the parameters of selectivity as a random process that extends throughout the forecast period, and to update estimated selectivity in each year of the forecast based upon the random realization of these parameters.

15. Example Input Files

All of the files below are **.ss_new* files as output by SS.

15.1 STARTER.SS

15.1 STARTER.SS

```
#V3.23b
#C starter comment here
simple.dat
simple.ctl
0 # 0=use init values in control file; 1=use ss3.par
1 # run display detail (0,1,2)
1 # detailed age-structured reports in REPORT.SSO (0,1)
0 # write detailed checkup.sso file (0,1)
0 # write parm values to ParmTrace.sso (0=no,1=good,active; 2=good,all; 3=every_iter,all_parms; 4=every,active)
1 # write to cumreport.sso (0=no,1=like&timeseries; 2=add survey fits)
1 # Include prior_like for non-estimated parameters (0,1)
1 # Use Soft Boundaries to aid convergence (0,1) (recommended)
3 # Number of datafiles to produce: 1st is input, 2nd is estimates, 3rd and higher are bootstrap
10 # Turn off estimation for parameters entering after this phase
10 # MCeval burn interval
2 # MCeval thin interval
0 # jitter initial parm value by this fraction
1969 # min yr for sdreport outputs (-1 for styr)
2011 # max yr for sdreport outputs (-1 for endyr; -2 for endyr+Nforecastyrs
0 # N individual STD years
#vector of year values

0.0001 # final convergence criteria (e.g. 1.0e-04)
0 # retrospective year relative to end year (e.g. -4)
1 # min age for calc of summary biomass
1 # Depletion basis:  denom is: 0=skip; 1=rel X*B0; 2=rel X*Bmsy; 3=rel X*B_styr
0.4 # Fraction (X) for Depletion denominator (e.g. 0.4)
1 # SPR_report_basis:  0=skip; 1=(1-SPR)/(1-SPR_tgt); 2=(1-SPR)/(1-SPR_MSY); 3=(1-SPR)/(1-SPR_Btarget); 4=rawSPR
4 # F_report_units: 0=skip; 1=exploitation(Bio); 2=exploitation(Num); 3=sum(Frates); 4=true F for range of ages
 20 23 #_min and max age over which average F will be calculated
1 # F_report_basis: 0=raw; 1=F/Fspr; 2=F/Fmsy ; 3=F/Fbtgt
999 # check value for end of file
```

15.2 RUNNUMBER.SS

```
7
```

15.3 PROFILEVALUES.SS

```
2      # number of parameters using profile feature
0.4    # value for first selected parameter when runnumber equals 1
11.0   # value for second selected parameter when runnumber equals 1
0.5    # value for first selected parameter when runnumber equals 2
11.0   # value for second selected parameter when runnumber equals 2
0.4    # value for first selected parameter when runnumber equals 3
13.0   # value for second selected parameter when runnumber equals 3
etc.
```

15.4 FORECAST.SS

```
#V3.23b
#C  generic forecast file
# for all year entries except rebuilder; enter either: actual year, -999 for styr, 0 for endyr, neg number for rel. endyr
1 # Benchmarks: 0=skip; 1=calc F_spr,F_btgt,F_msy
2 # MSY: 1= set to F(SPR); 2=calc F(MSY); 3=set to F(Btgt); 4=set to F(endyr)
0.4 # SPR target (e.g. 0.40)
0.342 # Biomass target (e.g. 0.40)
#_Bmark_years: beg_bio, end_bio, beg_selex, end_selex, beg_relF, end_relF (enter actual year, or values of 0 or -integer to be rel. endyr)
 0 0 0 0 0 0
#  2001 2001 2001 2001 2001 2001 # after processing
1 #Bmark_relF_Basis: 1 = use year range; 2 = set relF same as forecast below
#
1 # Forecast: 0=none; 1=F(SPR); 2=F(MSY) 3=F(Btgt); 4=Ave F (uses first-last relF yrs); 5=input annual F scalar
10 # N forecast years
0.2 # F scalar (only used for Do_Forecast==5)
#_Fcast_years:  beg_selex, end_selex, beg_relF, end_relF  (enter actual year, or values of 0 or -integer to be rel. endyr)
 0 0 -10 0
#  2001 2001 1991 2001 # after processing
1 # Control rule method (1=catch=f(SSB) west coast; 2=F=f(SSB) )
0.4 # Control rule Biomass level for constant F (as frac of Bzero, e.g. 0.40); (Must be > the no F level below)
```

```
0.1 # Control rule Biomass level for no F (as frac of Bzero, e.g. 0.10)
0.75 # Control rule target as fraction of Flimit (e.g. 0.75)
3 #_N forecast loops (1=OFL only; 2=ABC; 3=get F from forecast ABC catch with allocations applied)
3 #_First forecast loop with stochastic recruitment
0 #_Forecast loop control #3 (reserved for future bells&whistles)
0 #_Forecast loop control #4 (reserved for future bells&whistles)
0 #_Forecast loop control #5 (reserved for future bells&whistles)
2010  #FirstYear for caps and allocations (should be after years with fixed inputs)
0 # stddev of log(realized catch/target catch) in forecast (set value>0.0 to cause active impl_error)
0 # Do West Coast gfish rebuilder output (0/1)
1999 # Rebuilder:  first year catch could have been set to zero (Ydecl)(-1 to set to 1999)
2002 # Rebuilder:  year for current age structure (Yinit) (-1 to set to endyear+1)
1 # fleet relative F:  1=use first-last alloc year; 2=read seas(row) x fleet(col) below
# Note that fleet allocation is used directly as average F if Do_Forecast=4
2 # basis for fcast catch tuning and for fcast catch caps and allocation  (2=deadbio; 3=retainbio; 5=deadnum; 6=retainnum)
# Conditional input if relative F choice = 2
# Fleet relative F:  rows are seasons, columns are fleets
#_Fleet:  FISHERY1
#  1
# max totalcatch by fleet (-1 to have no max) must enter value for each fleet
 -1
# max totalcatch by area (-1 to have no max); must enter value for each fleet
 -1
# fleet assignment to allocation group (enter group ID# for each fleet, 0 for not included in an alloc group)
 0
#_Conditional on >1 allocation group
# allocation fraction for each of: 0 allocation groups
# no allocation groups
0 # Number of forecast catch levels to input (else calc catch from forecast F)
2 # basis for input Fcast catch:  2=dead catch; 3=retained catch; 99=input Hrate(F) (units are from fleetunits; note new codes in SSV3.20)
# Input fixed catch values
#Year Seas Fleet Catch(or_F)

#
999 # verify end of input
```

15.5 CONTROL FILE

```
#V3.23b
#C growth parameters are estimated
#C spawner-recruitment bias adjustment Not tuned For optimality
#_data_and_control_files: simple.dat // simple.ctl
#_SS-V3.21d-safe;_06/09/2011;_Stock_Synthesis_by_Richard_Methot_(NOAA)_using_ADMB_10
1  #_N_Growth_Patterns
1 #_N_Morphs_Within_GrowthPattern
#_Cond 1 #_Morph_between/within_stdev_ratio (no read if N_morphs=1)
#_Cond  1 #vector_Morphdist_(-1_in_first_val_gives_normal_approx)
#
#_Cond 0  #  N recruitment designs goes here if N_GP*nseas*area>1
#_Cond 0  #  placeholder for recruitment interaction request
#_Cond 1 1 1  # example recruitment design element for GP=1, seas=1, area=1
#
#_Cond 0 # N_movement_definitions goes here if N_areas > 1
#_Cond 1.0 # first age that moves (real age at begin of season, not integer) also cond on do_migration>0
#_Cond 1 1 1 2 4 10 # example move definition for seas=1, morph=1, source=1 dest=2, age1=4, age2=10
#
0 #_Nblock_Patterns
#_Cond 0 #_blocks_per_pattern
# begin and end years of blocks
#
0.5 #_fracfemale
0 #_natM_type:_0=1Parm; 1=N_breakpoints;_2=Lorenzen;_3=agespecific;_4=agespec_withseasinterpolate
  #_no additional input for selected M option; read 1P per morph
1 # GrowthModel: 1=vonBert with L1&L2; 2=Richards with L1&L2; 3=not implemented; 4=not implemented
0 #_Growth_Age_for_L1
25 #_Growth_Age_for_L2 (999 to use as Linf)
0 #_SD_add_to_LAA (set to 0.1 for SS2 V1.x compatibility)
0 #_CV_Growth_Pattern:  0 CV=f(LAA); 1 CV=F(A); 2 SD=F(LAA); 3 SD=F(A); 4 logSD=F(A)
1 #_maturity_option:   1=length logistic; 2=age logistic; 3=read age-maturity matrix by growth_pattern; 4=read age-
fecundity; 5=read fec and wt from wtatage.ss
#_placeholder for empirical age-maturity by growth pattern
1 #_First_Mature_Age
1 #_fecundity option:(1)eggs=Wt*(a+b*Wt);(2)eggs=a*L^b;(3)eggs=a*Wt^b; (4)eggs=a+b*L; (5)eggs=a+b*W
0 #_hermaphroditism option:  0=none; 1=age-specific fxn
1 #_parameter_offset_approach (1=none, 2= M, G, CV_G as offset from female-GP1, 3=like SS2 V1.x)
2 #_env/block/dev_adjust_method (1=standard; 2=logistic transform keeps in base parm bounds; 3=standard w/ no bound
check)
#
#_growth_parms
```

```
#_LO HI INIT PRIOR PR_type SD PHASE env-var use_dev dev_minyr dev_maxyr dev_stddev Block Block_Fxn
 0.05 0.15 0.1 0.1 -1 0.8 -3 0 0 0 0 0 0 0 0 # NatM_p_1_Fem_GP_1
 -10 45 21.6552 36 0 10 2 0 0 0 0 0 0 0 0 # L_at_Amin_Fem_GP_1
 40 90 71.6492 70 0 10 4 0 0 0 0 0 0 0 0 # L_at_Amax_Fem_GP_1
 0.05 0.25 0.147282 0.15 0 0.8 4 0 0 0 0 0 0 0 0 # VonBert_K_Fem_GP_1
 0.05 0.25 0.1 0.1 -1 0.8 -3 0 0 0 0 0 0 0 0 # CV_young_Fem_GP_1
 0.05 0.25 0.1 0.1 -1 0.8 -3 0 0 0 0 0 0 0 0 # CV_old_Fem_GP_1
 0.05 0.15 0.1 0.1 -1 0.8 -3 0 0 0 0 0 0 0 0 # NatM_p_1_Mal_GP_1
 1 45 0 36 -1 10 -3 0 0 0 0 0 0 0 0 # L_at_Amin_Mal_GP_1
 40 90 69.5361 70 0 10 4 0 0 0 0 0 0 0 0 # L_at_Amax_Mal_GP_1
 0.05 0.25 0.163516 0.15 0 0.8 4 0 0 0 0 0 0 0 0 # VonBert_K_Mal_GP_1
 0.05 0.25 0.1 0.1 -1 0.8 -3 0 0 0 0 0 0 0 0 # CV_young_Mal_GP_1
 0.05 0.25 0.1 0.1 -1 0.8 -3 0 0 0 0 0 0 0 0 # CV_old_Mal_GP_1
 -3 3 2.44e-006 2.44e-006 -1 0.8 -3 0 0 0 0 0 0 0 0 # Wtlen_1_Fem
 -3 4 3.34694 3.34694 -1 0.8 -3 0 0 0 0 0 0 0 0 # Wtlen_2_Fem
 50 60 55 55 -1 0.8 -3 0 0 0 0 0 0 0 0 # Mat50%_Fem
 -3 3 -0.25 -0.25 -1 0.8 -3 0 0 0 0 0 0 0 0 # Mat_slope_Fem
 -3 3 1 1 -1 0.8 -3 0 0 0 0 0 0 0 0 # Eggs/kg_inter_Fem
 -3 3 0 0 -1 0.8 -3 0 0 0 0 0 0 0 0 # Eggs/kg_slope_wt_Fem
 -3 3 2.44e-006 2.44e-006 -1 0.8 -3 0 0 0 0 0 0 0 0 # Wtlen_1_Mal
 -3 4 3.34694 3.34694 -1 0.8 -3 0 0 0 0 0 0 0 0 # Wtlen_2_Mal
 0 0 0 0 -1 0 -4 0 0 0 0 0 0 0 0 # RecrDist_GP_1
 0 0 0 0 -1 0 -4 0 0 0 0 0 0 0 0 # RecrDist_Area_1
 0 0 0 0 -1 0 -4 0 0 0 0 0 0 0 0 # RecrDist_Seas_1
 0 0 0 0 -1 0 -4 0 0 0 0 0 0 0 0 # CohortGrowDev
#
#_Cond 0  #custom_MG-env_setup (0/1)
#_Cond -2 2 0 0 -1 99 -2 #_placeholder when no MG-environ parameters
#
#_Cond 0  #custom_MG-block_setup (0/1)
#_Cond -2 2 0 0 -1 99 -2 #_placeholder when no MG-block parameters
#_Cond No MG parm trends
#
#_seasonal_effects_on_biology_parms
 0 0 0 0 0 0 0 0 0 0 #_femwtlen1,femwtlen2,mat1,mat2,fec1,fec2,Malewtlen1,malewtlen2,L1,K
#_Cond -2 2 0 0 -1 99 -2 #_placeholder when no seasonal MG parameters
#
#_Cond -4 #_MGparm_Dev_Phase
#
#_Spawner-Recruitment
3 #_SR_function: 2=Ricker; 3=std_B-H; 4=SCAA; 5=Hockey; 6=B-H_flattop; 7=survival_3Parm
#_LO HI INIT PRIOR PR_type SD PHASE
```

```
 3 31 8.81544 10.3 -1 10 1 # SR_LN(R0)
 0.2 1 0.613717 0.7 1 0.05 4 # SR_BH_steep
 0 2 0.6 0.8 -1 0.8 -4 # SR_sigmaR
 -5 5 0.1 0 -1 1 -3 # SR_envlink
 -5 5 0 0 -1 1 -4 # SR_R1_offset
 0 0 0 0 -1 0 -99 # SR_autocorr
0 #_SR_env_link
0 #_SR_env_target_0=none;1=devs;_2=R0;_3=steepness
1 #do_recdev:  0=none; 1=devvector; 2=simple deviations
1971 # first year of main recr_devs; early devs can preceed this era
2001 # last year of main recr_devs; forecast devs start in following year
2 #_recdev phase
1 # (0/1) to read 13 advanced options
 0 #_recdev_early_start (0=none; neg value makes relative to recdev_start)
 -4 #_recdev_early_phase
 0 #_forecast_recruitment phase (incl. late recr) (0 value resets to maxphase+1)
 1 #_lambda for Fcast_recr_like occurring before endyr+1
 1900 #_last_early_yr_nobias_adj_in_MPD
 1900 #_first_yr_fullbias_adj_in_MPD
 2001 #_last_yr_fullbias_adj_in_MPD
 2002 #_first_recent_yr_nobias_adj_in_MPD
 1 #_max_bias_adj_in_MPD (-1 to override ramp and set biasadj=1.0 for all estimated recdevs)
 0 #_period of cycles in recruitment (N parms read below)
 -5 #min rec_dev
 5 #max rec_dev
 0 #_read_recdevs
#_end of advanced SR options
#
#_placeholder for full parameter lines for recruitment cycles
# read specified recr devs
#_Yr Input_value
#
# all recruitment deviations
#DisplayOnly 0.127813 # Main_RecrDev_1971
#DisplayOnly -0.0629072 # Main_RecrDev_1972
#DisplayOnly 0.0999946 # Main_RecrDev_1973
#DisplayOnly -0.173914 # Main_RecrDev_1974
#DisplayOnly 0.0306906 # Main_RecrDev_1975
#DisplayOnly 0.714754 # Main_RecrDev_1976
#DisplayOnly -0.0229372 # Main_RecrDev_1977
#DisplayOnly 0.00347081 # Main_RecrDev_1978
#DisplayOnly 0.260891 # Main_RecrDev_1979
```

```
#DisplayOnly 0.173281 # Main_RecrDev_1980
#DisplayOnly 0.0891999 # Main_RecrDev_1981
#DisplayOnly -0.227374 # Main_RecrDev_1982
#DisplayOnly -0.440643 # Main_RecrDev_1983
#DisplayOnly -0.312905 # Main_RecrDev_1984
#DisplayOnly 0.391936 # Main_RecrDev_1985
#DisplayOnly 0.551136 # Main_RecrDev_1986
#DisplayOnly 0.218287 # Main_RecrDev_1987
#DisplayOnly 0.15476 # Main_RecrDev_1988
#DisplayOnly -0.384699 # Main_RecrDev_1989
#DisplayOnly 0.596713 # Main_RecrDev_1990
#DisplayOnly -0.68218 # Main_RecrDev_1991
#DisplayOnly -0.273103 # Main_RecrDev_1992
#DisplayOnly -0.829262 # Main_RecrDev_1993
#DisplayOnly 0.365425 # Main_RecrDev_1994
#DisplayOnly -0.604348 # Main_RecrDev_1995
#DisplayOnly 0.455566 # Main_RecrDev_1996
#DisplayOnly 1.11144 # Main_RecrDev_1997
#DisplayOnly -0.545922 # Main_RecrDev_1998
#DisplayOnly -0.65609 # Main_RecrDev_1999
#DisplayOnly 0.172111 # Main_RecrDev_2000
#DisplayOnly -0.301188 # Main_RecrDev_2001
#DisplayOnly 0 # ForeRecr_2002
#DisplayOnly 0 # ForeRecr_2003
#DisplayOnly 0 # ForeRecr_2004
#DisplayOnly 0 # ForeRecr_2005
#DisplayOnly 0 # ForeRecr_2006
#DisplayOnly 0 # ForeRecr_2007
#DisplayOnly 0 # ForeRecr_2008
#DisplayOnly 0 # ForeRecr_2009
#DisplayOnly 0 # ForeRecr_2010
#DisplayOnly 0 # ForeRecr_2011
#DisplayOnly 0 # Impl_err_2002
#DisplayOnly 0 # Impl_err_2003
#DisplayOnly 0 # Impl_err_2004
#DisplayOnly 0 # Impl_err_2005
#DisplayOnly 0 # Impl_err_2006
#DisplayOnly 0 # Impl_err_2007
#DisplayOnly 0 # Impl_err_2008
#DisplayOnly 0 # Impl_err_2009
#DisplayOnly 0 # Impl_err_2010
#DisplayOnly 0 # Impl_err_2011
```

```
#
#Fishing Mortality info
0.3 # F ballpark for tuning early phases
-2001 # F ballpark year (neg value to disable)
3 # F_Method:  1=Pope; 2=instan. F; 3=hybrid (hybrid is recommended)
2.9 # max F or harvest rate, depends on F_Method
# no additional F input needed for Fmethod 1
# if Fmethod=2; read overall start F value; overall phase; N detailed inputs to read
# if Fmethod=3; read N iterations for tuning for Fmethod 3
4  # N iterations for tuning F in hybrid method (recommend 3 to 7)
#
#_initial_F_parms
#_LO HI INIT PRIOR PR_type SD PHASE
 0 1 0 0.01 0 99 -1 # InitF_1FISHERY1
#
#_Q_setup
 #  Q_type  options:   <0=mirror,  0=median_float,  1=mean_float,  2=parameter,  3=parm_w_random_dev,  4=parm_w_randwalk,
5=mean_unbiased_float_assign_to_parm
 #_Den-dep  env-var  extra_se  Q_type
 0 0 0 0 # 1 FISHERY1
 0 0 1 2 # 2 SURVEY1
 0 0 0 2 # 3 SURVEY2
#
#_Cond 0 #_If q has random component, then 0=read one parm for each fleet with random q; 1=read a parm for each year of
index
#_Q_parms(if_any)
# LO HI INIT PRIOR PR_type SD PHASE
 0 0.5 0 0.05 1 0 -4 # Q_extraSD_2_SURVEY1
 -7 5 0.515263 0 -1 1 1 # Q_base_2_SURVEY1
 -7 5 -6.62828 0 -1 1 1 # Q_base_3_SURVEY2
#
#_size_selex_types
#_Pattern Discard Male Special
 1 0 0 0 # 1 FISHERY1
 1 0 0 0 # 2 SURVEY1
 0 0 0 0 # 3 SURVEY2
#
#_age_selex_types
#_Pattern ___ Male Special
 11 0 0 0 # 1 FISHERY1
 11 0 0 0 # 2 SURVEY1
 11 0 0 0 # 3 SURVEY2
```

```
#_LO HI INIT PRIOR PR_type SD PHASE env-var use_dev dev_minyr dev_maxyr dev_stddev Block Block_Fxn
 19 80 53.6526 50 1 0.01 2 0 0 0 0 0 0 0 # SizeSel_1P_1_FISHERY1
 0.01 60 18.9204 15 1 0.01 3 0 0 0 0 0 0 0 # SizeSel_1P_2_FISHERY1
 19 70 36.6499 30 1 0.01 2 0 0 0 0 0 0 0 # SizeSel_2P_1_SURVEY1
 0.01 60 6.58921 10 1 0.01 3 0 0 0 0 0 0 0 # SizeSel_2P_2_SURVEY1
 0 40 0 5 -1 99 -1 0 0 0 0 0 0 0 # AgeSel_1P_1_FISHERY1
 0 40 40 6 -1 99 -1 0 0 0 0 0 0 0 # AgeSel_1P_2_FISHERY1
 0 40 0 5 -1 99 -1 0 0 0 0 0 0 0 # AgeSel_2P_1_SURVEY1
 0 40 40 6 -1 99 -1 0 0 0 0 0 0 0 # AgeSel_2P_2_SURVEY1
 0 40 0 5 -1 99 -1 0 0 0 0 0 0 0 # AgeSel_3P_1_SURVEY2
 0 40 0 6 -1 99 -1 0 0 0 0 0 0 0 # AgeSel_3P_2_SURVEY2
#_Cond 0 #_custom_sel-env_setup (0/1)
#_Cond -2 2 0 0 -1 99 -2 #_placeholder when no enviro fxns
#_Cond 0 #_custom_sel-blk_setup (0/1)
#_Cond -2 2 0 0 -1 99 -2 #_placeholder when no block usage
#_Cond No selex parm trends
#_Cond -4 # placeholder for selparm_Dev_Phase
#_Cond 0 #_env/block/dev_adjust_method (1=standard; 2=logistic trans to keep in base parm bounds; 3=standard w/ no
bound check)
#
# Tag loss and Tag reporting parameters go next
0  # TG_custom:  0=no read; 1=read if tags exist
#_Cond -6 6 1 1 2 0.01 -4 0 0 0 0 0 0 0  #_placeholder if no parameters
#
1 #_Variance_adjustments_to_input_values
#_fleet: 1 2 3
  0 0 0 #_add_to_survey_CV
  0 0 0 #_add_to_discard_stddev
  0 0 0 #_add_to_bodywt_CV
  1 1 1 #_mult_by_lencomp_N
  1 1 1 #_mult_by_agecomp_N
  1 1 1 #_mult_by_size-at-age_N
#
4 #_maxlambdaphase
1 #_sd_offset
#
3 # number of changes to make to default Lambdas (default value is 1.0)
# Like_comp codes:  1=surv; 2=disc; 3=mnwt; 4=length; 5=age; 6=SizeFreq; 7=sizeage; 8=catch;
# 9=init_equ_catch; 10=recrdev; 11=parm_prior; 12=parm_dev; 13=CrashPen; 14=Morphcomp; 15=Tag-comp; 16=Tag-negbin
#like_comp fleet/survey  phase  value  sizefreq_method
 1 2 2 1 1
 4 2 2 1 1
```

```
 4 2 3 1 1
#
# lambdas (for info only; columns are phases)
#  0 0 0 0 #_CPUE/survey:_1
#  1 1 1 1 #_CPUE/survey:_2
#  1 1 1 1 #_CPUE/survey:_3
#  1 1 1 1 #_lencomp:_1
#  1 1 1 1 #_lencomp:_2
#  0 0 0 0 #_lencomp:_3
#  1 1 1 1 #_agecomp:_1
#  1 1 1 1 #_agecomp:_2
#  0 0 0 0 #_agecomp:_3
#  1 1 1 1 #_size-age:_1
#  1 1 1 1 #_size-age:_2
#  0 0 0 0 #_size-age:_3
#  1 1 1 1 #_init_equ_catch
#  1 1 1 1 #_recruitments
#  1 1 1 1 #_parameter-priors
#  1 1 1 1 #_parameter-dev-vectors
#  1 1 1 1 #_crashPenLambda
1 # (0/1) read specs for more stddev reporting
 1 1 -1 5 1 5 1 -1 5 # selex type, len/age, year, N selex bins, Growth pattern, N growth ages, NatAge_area(-1 for all),
NatAge_yr, N Natages
 5 15 25 35 43 # vector with selex std bin picks (-1 in first bin to self-generate)
 1 2 14 26 40 # vector with growth std bin picks (-1 in first bin to self-generate)
 1 2 14 26 40 # vector with NatAge std bin picks (-1 in first bin to self-generate)
999
```

15.6 DATA FILE

```
#V3.23b
#C data file for simple example
1971 #_styr
2001 #_endyr
1 #_nseas
 12 #_months/season
1 #_spawn_seas
1 #_Nfleet
2 #_Nsurveys
1 #_N_areas
FISHERY1%SURVEY1%SURVEY2
 0.5 0.5 0.5 #_surveytiming_in_season
 1 1 1 #_area_assignments_for_each_fishery_and_survey
 1 #_units of catch:  1=bio; 2=num
 0.01 #_se of log(catch) only used for init_eq_catch and for Fmethod 2 and 3
2 #_Ngenders
40 #_Nages
 0 #_init_equil_catch_for_each_fishery
31 #_N_lines_of_catch_to_read
#_catch_biomass(mtons):_columns_are_fisheries,year,season
 0 1971 1
 200 1972 1
 1000 1973 1
 1000 1974 1
 2000 1975 1
 3000 1976 1
 4000 1977 1
 5000 1978 1
 6000 1979 1
 8000 1980 1
 10000 1981 1
 10000 1982 1
 10000 1983 1
 10000 1984 1
 10000 1985 1
 10000 1986 1
 10000 1987 1
 9000 1988 1
 8000 1989 1
 7000 1990 1
 6000 1991 1
```

```
 4000 1992 1
 4000 1993 1
 4000 1994 1
 4000 1995 1
 4000 1996 1
 3000 1997 1
 3000 1998 1
 3000 1999 1
 3000 2000 1
 3000 2001 1
#
#
21 #_N_cpue_and_surveyabundance_observations
#_Units:  0=numbers; 1=biomass; 2=F
#_Errtype:  -1=normal; 0=lognormal; >0=T
#_Fleet Units Errtype
1 1 0 # FISHERY1
2 1 0 # SURVEY1
3 0 0 # SURVEY2
#_year seas index obs err
 1977 1 2 339689 0.3 # SURVEY1
 1980 1 2 193353 0.3 # SURVEY1
 1983 1 2 151984 0.3 # SURVEY1
 1986 1 2 55221.8 0.3 # SURVEY1
 1989 1 2 59232.3 0.3 # SURVEY1
 1992 1 2 31137.5 0.3 # SURVEY1
 1995 1 2 35845.4 0.3 # SURVEY1
 1998 1 2 27492.6 0.3 # SURVEY1
 2001 1 2 37338.3 0.3 # SURVEY1
 1990 1 3 5.19333 0.7 # SURVEY2
 1991 1 3 1.1784 0.7 # SURVEY2
 1992 1 3 5.94383 0.7 # SURVEY2
 1993 1 3 0.770106 0.7 # SURVEY2
 1994 1 3 16.318 0.7 # SURVEY2
 1995 1 3 1.36339 0.7 # SURVEY2
 1996 1 3 4.76482 0.7 # SURVEY2
 1997 1 3 51.0707 0.7 # SURVEY2
 1998 1 3 1.36095 0.7 # SURVEY2
 1999 1 3 0.862531 0.7 # SURVEY2
 2000 1 3 5.97125 0.7 # SURVEY2
 2001 1 3 1.69379 0.7 # SURVEY2
#
```

```
0 #_N_fleets_with_discard
#_discard_units (1=same_as_catchunits(bio/num); 2=fraction; 3=numbers)
#_discard_errtype:  >0 for DF of T-dist(read CV below); 0 for normal with CV; -1 for normal with se; -2 for lognormal
#Fleet Disc_units err_type
0 #N discard obs
#_year seas index obs err
#
0 #_N_meanbodywt_obs
30 #_DF_for_meanbodywt_T-distribution_like
#
2 # length bin method: 1=use databins; 2=generate from binwidth,min,max below; 3=read vector
2 # binwidth for population size comp
10 # minimum size in the population (lower edge of first bin and size at age 0.00)
94 # maximum size in the population (lower edge of last bin)
#
0 #_comp_tail_compression
1e-007 #_add_to_comp
0 #_combine males into females at or below this bin number
25 #_N_LengthBins
 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 68 72 76 80 90
40 #_N_Length_obs
#Yr Seas Flt/Svy Gender Part Nsamp datavector(female-male)
 1971 1 1 3 0 125 0 0 0 0 0 0 0 0 0 4 1 1 2 4 1 5 6 2 3 11 8 4 5 0 0 0 0 0 0 0 0 0 0 1 0 1 3 0 3 4 2 4 5 9 17 8 3 8 0 0
 1972 1 1 3 0 125 0 0 0 0 0 0 0 0 0 3 0 1 2 1 1 6 2 7 4 10 10 4 5 3 0 0 0 0 0 0 0 0 0 1 3 2 4 1 3 1 4 4 7 3 8 11 4 10 0
0
 1973 1 1 3 0 125 0 0 0 0 0 0 0 0 0 0 0 7 3 4 5 6 3 10 12 6 10 9 0 0 0 0 0 0 0 0 0 0 0 0 3 0 1 3 0 7 2 6 7 8 5 5 3
0
 1974 1 1 3 0 125 0 0 0 0 0 0 0 0 0 2 2 0 1 1 1 4 5 3 8 8 10 4 7 0 0 0 0 0 0 0 0 0 1 2 0 4 0 0 1 5 6 6 4 6 15 11 5 0 3
0
 1975 1 1 3 0 125 0 0 0 0 0 0 0 2 1 2 1 1 3 0 2 5 6 2 3 5 9 10 10 0 0 0 0 0 0 0 0 0 0 0 4 2 2 1 2 3 5 1 4 5 13 11 6 4 0
0
 1976 1 1 3 0 125 0 0 0 0 0 0 0 2 1 0 2 2 0 3 2 3 3 3 7 18 14 4 2 2 0 0 0 0 0 0 0 1 0 0 0 0 0 1 2 4 6 6 5 7 12 6 4 3 0
0
 1977 1 1 3 0 125 0 0 0 0 0 0 0 1 0 2 0 2 2 4 0 2 6 7 5 11 7 8 5 4 0 0 0 0 0 0 2 1 3 0 1 3 3 2 0 1 4 5 3 7 7 9 5 3 0 0
 1978 1 1 3 0 125 0 0 0 0 0 0 5 1 1 1 0 1 3 1 8 4 4 6 5 9 8 3 6 5 0 0 0 0 0 0 0 0 2 1 1 2 1 1 2 2 4 1 4 1 13 9 6 4 0 0
 1979 1 1 3 0 125 0 0 0 0 0 0 0 0 0 0 3 5 2 1 5 0 5 5 2 7 4 7 5 5 0 0 0 0 0 0 0 0 0 2 2 1 3 2 7 2 4 4 5 8 10 8 6 4 1 0
 1980 1 1 3 0 125 0 0 0 0 0 0 0 4 0 0 1 0 2 4 3 2 3 2 3 16 11 12 4 2 0 0 0 0 0 0 0 0 0 1 4 1 1 2 3 5 2 6 3 1 10 11 4 2
0 0
 1981 1 1 3 0 125 0 0 0 0 0 0 1 0 0 0 3 1 2 2 4 5 2 7 3 13 9 8 4 0 0 0 0 0 0 0 2 1 1 1 2 2 3 3 1 6 1 2 1 7 5 10 6 7 0 0
 1982 1 1 3 0 125 0 0 0 0 0 0 0 0 0 5 2 1 3 2 3 8 2 5 4 4 6 10 11 0 0 0 0 0 0 0 0 0 0 1 0 3 0 2 1 5 6 1 8 5 5 10 5 2 5 0
0
 1983 1 1 3 0 125 0 0 0 0 0 0 0 0 0 0 7 1 1 5 4 2 2 6 2 8 13 8 6 0 0 0 0 0 0 0 0 0 0 0 0 4 1 0 3 3 0 4 9 5 4 7 8 6 6 0 0
```

```
 1984 1 1 3 0 125 0 0 0 0 0 0 1 0 0 4 3 0 3 1 2 5 2 4 7 11 9 6 8 0 0 0 0 0 0 0 0 0 3 3 1 1 3 3 3 2 2 4 4 8 11 4 5 2 0 0
 1985 1 1 3 0 125 0 0 0 0 0 0 0 0 1 1 2 2 5 0 3 3 5 11 4 8 9 3 2 4 0 0 0 0 0 0 0 0 1 0 1 2 0 3 8 3 4 3 8 4 13 7 4 1 0 0
 1986 1 1 3 0 125 0 0 0 3 1 0 1 2 0 4 2 0 0 4 2 8 3 5 11 5 6 6 1 0 0 0 0 0 0 0 2 2 0 1 2 1 3 4 2 3 4 6 5 5 6 4 6 5 0 0
 1987 1 1 3 0 125 0 0 0 0 1 1 1 1 1 0 2 1 6 4 2 7 6 3 5 11 9 5 4 0 0 0 0 0 0 0 0 0 2 1 0 5 2 4 3 4 4 4 2 4 7 6 5 2 0 0
 1988 1 1 3 0 125 0 0 0 0 0 2 0 1 4 2 1 1 2 2 1 7 4 5 6 9 9 2 1 0 0 0 0 0 0 0 2 1 1 3 1 3 6 3 3 0 4 5 3 5 9 9 8 0 0 0
 1989 1 1 3 0 125 0 0 0 0 0 1 0 2 1 3 3 2 1 4 4 3 4 2 3 9 5 11 2 0 0 0 0 0 0 0 0 3 6 2 1 3 0 4 3 3 2 5 7 7 9 3 3 4 0 0
 1990 1 1 3 0 125 0 0 0 0 0 0 0 2 2 2 2 2 2 2 9 4 4 6 6 8 4 4 1 0 0 0 0 0 0 0 1 1 2 2 3 8 2 8 6 6 3 2 3 4 6 5 1 2 0 0
 1991 1 1 3 0 125 0 0 0 0 0 0 0 3 0 3 3 5 5 4 3 3 0 1 6 10 4 4 0 0 0 0 0 0 0 1 1 1 1 3 4 6 5 3 5 6 6 6 6 4 7 3 3 0 0 0
 1992 1 1 3 0 125 0 0 0 0 2 2 0 1 1 1 3 3 2 7 6 4 4 2 5 6 3 6 0 0 0 0 0 0 0 0 0 5 3 1 3 5 3 5 8 3 4 6 3 13 4 1 0 0 0
 1993 1 1 3 0 125 0 0 0 0 0 0 1 2 2 2 2 2 4 5 10 5 7 3 2 12 7 6 0 0 0 0 0 0 0 0 0 0 3 1 1 3 2 6 4 8 4 6 4 2 4 3 1 1 0 0
 1994 1 1 3 0 125 0 0 0 0 0 0 0 0 0 4 1 4 3 4 4 9 4 6 7 8 5 3 2 0 0 0 0 0 0 0 0 0 2 0 2 1 1 4 4 10 5 8 6 3 5 6 1 3 0 0
 1995 1 1 3 0 125 0 0 0 1 0 0 1 1 1 1 2 2 5 8 4 11 5 5 4 8 7 0 0 0 0 0 0 0 0 1 0 0 1 1 3 3 1 2 6 3 4 4 8 3 12 4 3 0 0 0
 1996 1 1 3 0 125 0 0 0 1 0 2 1 0 2 4 3 3 2 3 6 6 3 3 4 11 6 6 0 0 0 0 0 0 0 0 0 1 2 0 3 3 1 0 5 4 6 7 4 5 10 3 4 1 0 0
 1997 1 1 3 0 125 0 0 0 2 0 0 2 2 0 0 3 1 6 4 6 2 9 4 5 9 12 0 0 0 0 0 0 0 0 0 3 1 0 5 3 2 4 1 1 6 4 1 6 6 5 6 4 0 0 0
 1998 1 1 3 0 125 0 0 0 0 3 1 2 2 2 2 3 1 3 6 2 0 7 4 5 12 3 1 2 0 0 0 0 0 0 4 1 1 0 2 2 0 1 1 4 6 2 5 4 6 13 7 4 1 0 0
 1999 1 1 3 0 125 0 0 0 0 1 0 1 1 3 0 1 2 2 8 3 4 7 3 5 6 5 7 0 0 0 0 0 0 0 0 0 7 3 4 2 3 2 5 2 11 3 5 1 5 7 4 2 0 0 0
 2000 1 1 3 0 125 0 0 0 0 0 1 0 0 1 2 4 3 1 6 4 4 3 3 4 5 11 0 0 0 0 0 0 0 0 0 2 4 4 3 3 6 3 4 1 8 3 5 1 4 11 1 5 5 0 0
 2001 1 1 3 0 125 0 0 0 0 2 1 0 1 1 0 2 7 6 9 4 2 5 6 4 7 6 4 0 0 0 0 0 0 0 0 2 0 1 0 2 3 2 5 3 8 3 3 5 2 10 6 3 0 0 0
 1977 1 2 3 0 125 0 0 0 0 3 0 0 2 2 3 1 2 5 0 5 6 5 3 3 8 4 10 0 0 0 0 0 0 0 0 0 6 3 3 2 2 5 2 3 3 8 1 1 6 5 8 3 2 0 0
 1980 1 2 3 0 125 0 0 0 0 1 1 1 3 2 2 1 3 6 1 2 5 1 3 3 8 3 3 4 1 0 0 0 0 0 1 1 2 3 4 4 4 4 4 1 1 1 5 3 5 14 7 5 2 0 0
 1983 1 2 3 0 125 0 0 0 0 2 3 3 5 2 4 5 2 3 2 5 5 6 5 3 3 1 8 0 0 0 0 0 0 0 2 2 1 2 2 4 2 6 2 3 5 2 4 4 1 6 10 0 0 0 0
 1986 1 2 3 0 125 0 0 0 0 2 1 1 4 6 2 3 1 1 1 5 5 5 3 3 7 7 3 2 0 0 0 0 0 1 2 1 3 2 1 5 0 2 5 6 7 3 5 2 3 7 4 4 0 0 0
 1989 1 2 3 0 125 0 0 0 0 0 5 8 3 3 5 1 2 4 1 2 2 4 3 2 3 3 2 0 0 0 0 0 0 2 2 3 5 2 5 8 8 7 3 2 4 3 6 3 1 8 0 0 0 0 0
 1992 1 2 3 0 125 0 0 0 0 0 5 6 6 5 3 2 5 6 6 5 5 1 3 1 3 4 0 0 0 0 0 0 0 0 0 2 4 3 6 5 3 6 6 2 5 4 3 1 3 1 2 3 0 0 0
 1995 1 2 3 0 125 0 0 0 0 2 0 0 4 7 5 5 5 6 2 5 6 5 6 0 3 4 1 0 0 0 0 0 0 0 2 3 0 1 2 1 5 3 4 9 5 3 3 4 2 5 4 3 0 0 0
 1998 1 2 3 0 125 0 0 0 3 1 1 2 3 4 6 4 6 5 3 1 2 1 1 1 5 2 2 0 0 0 0 0 0 0 10 5 4 2 3 7 2 1 4 4 5 3 2 3 1 8 6 2 0 0 0
 2001 1 2 3 0 125 0 0 0 0 0 2 3 5 7 5 9 2 9 5 4 4 1 1 2 2 8 0 0 0 0 0 0 0 0 2 1 4 6 5 6 4 3 4 4 5 1 3 2 1 3 2 0 0 0 0
#
17 #_N_age_bins
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 20 25
2 #_N_ageerror_definitions
 0.5 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5 10.5 11.5 12.5 13.5 14.5 15.5 16.5 17.5 18.5 19.5 20.5 21.5 22.5 23.5 24.5
25.5 26.5 27.5 28.5 29.5 30.5 31.5 32.5 33.5 34.5 35.5 36.5 37.5 38.5 39.5 40.5
 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001
0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001
0.001 0.001
 0.5 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5 10.5 11.5 12.5 13.5 14.5 15.5 16.5 17.5 18.5 19.5 20.5 21.5 22.5 23.5 24.5
25.5 26.5 27.5 28.5 29.5 30.5 31.5 32.5 33.5 34.5 35.5 36.5 37.5 38.5 39.5 40.5
 0.5 0.65 0.67 0.7 0.73 0.76 0.8 0.84 0.88 0.92 0.97 1.03 1.09 1.16 1.23 1.32 1.41 1.51 1.62 1.75 1.89 2.05 2.23 2.45
2.71 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
40 #_N_Agecomp_obs
1 #_Lbin_method: 1=poplenbins; 2=datalenbins; 3=lengths
```

```
1 #_combine males into females at or below this bin number
#Yr Seas Flt/Svy Gender Part Ageerr Lbin_lo Lbin_hi Nsamp datavector(female-male)
 1971 1 1 3 0 2 1 -1 75 0 0 0 0 3 1 1 4 2 1 0 1 2 2 13 2 3 0 0 4 2 1 1 2 1 2 2 1 2 1 2 6 5 8
 1972 1 1 3 0 2 1 -1 75 2 1 1 1 0 3 1 2 2 5 3 1 2 2 9 8 3 0 0 1 2 3 1 3 0 5 1 3 0 2 1 2 3 2
 1973 1 1 3 0 2 1 -1 75 0 0 1 0 1 1 2 3 3 1 1 5 2 2 7 4 3 0 0 0 4 1 3 5 1 2 3 1 3 2 0 5 3 6
 1974 1 1 3 0 2 1 -1 75 0 0 2 0 1 4 2 2 2 4 1 1 1 2 6 6 6 0 0 4 1 2 2 1 2 0 0 1 2 1 1 6 5 7
 1975 1 1 3 0 2 1 -1 75 0 0 1 2 3 1 1 1 2 1 2 2 2 3 10 3 4 0 0 0 0 10 1 2 3 2 1 0 0 0 0 9 3 6
 1976 1 1 3 0 2 1 -1 75 0 0 1 0 2 2 2 1 3 1 2 3 1 1 7 1 3 0 0 0 0 7 4 3 2 1 2 4 4 0 0 8 10 0
 1977 1 1 3 0 2 1 -1 75 0 0 0 0 7 1 0 0 2 4 2 2 3 1 7 2 3 0 0 2 1 4 2 3 3 4 2 2 2 0 1 8 3 4
 1978 1 1 3 0 2 1 -1 75 0 0 3 2 1 1 0 2 0 2 4 3 1 0 9 4 6 0 0 2 2 5 1 0 2 3 2 4 2 0 4 4 3 3
 1979 1 1 3 0 2 1 -1 75 2 0 1 5 2 1 2 3 3 3 2 2 1 0 3 7 0 0 0 2 0 1 0 2 3 2 5 1 3 1 2 6 9 1
 1980 1 1 3 0 2 1 -1 75 0 1 0 2 0 1 1 2 2 3 2 1 1 0 7 8 0 0 0 0 3 2 1 1 1 2 2 4 2 2 2 11 3 8
 1981 1 1 3 0 2 1 -1 75 0 4 0 3 7 2 2 2 2 1 1 2 2 1 4 4 6 0 0 3 2 2 1 1 3 2 2 0 1 2 2 5 3 3
 1982 1 1 3 0 2 1 -1 75 0 2 1 1 3 3 2 1 1 2 2 1 0 2 6 3 9 0 0 0 0 3 5 0 1 4 1 1 1 2 1 8 9 0
 1983 1 1 3 0 2 1 -1 75 0 0 0 6 1 2 2 2 1 1 4 5 0 0 6 2 7 0 0 3 1 3 5 1 0 1 1 3 0 3 3 5 3 4
 1984 1 1 3 0 2 1 -1 75 0 0 0 3 4 0 3 6 3 1 4 0 2 0 7 2 3 0 0 3 1 5 4 2 3 5 1 2 1 2 0 1 2 5
 1985 1 1 3 0 2 1 -1 75 0 0 0 5 1 2 4 5 0 2 4 3 2 3 3 4 5 0 0 0 1 2 3 2 4 2 0 2 3 1 1 7 2 2
 1986 1 1 3 0 2 1 -1 75 0 2 2 1 3 7 4 3 2 2 2 2 2 0 4 2 2 0 0 0 0 4 4 4 1 2 3 4 0 0 1 5 7 0
 1987 1 1 3 0 2 1 -1 75 0 3 1 3 1 2 3 4 2 3 3 2 2 1 3 2 0 0 0 7 1 5 1 4 2 4 3 2 3 1 0 2 1 4
 1988 1 1 3 0 2 1 -1 75 1 0 5 0 2 3 3 3 4 3 3 1 0 3 3 5 0 0 1 3 3 2 2 1 4 3 2 1 2 4 0 5 3 0
 1989 1 1 3 0 2 1 -1 75 0 3 1 1 4 3 7 1 5 1 1 4 1 0 1 7 0 0 0 5 3 4 1 1 5 3 1 5 2 1 0 2 2 0
 1990 1 1 3 0 2 1 -1 75 0 0 7 3 7 3 0 1 3 0 1 1 1 1 3 4 0 0 1 0 8 4 3 3 2 4 5 1 5 1 0 1 2 0
 1991 1 1 3 0 2 1 -1 75 0 0 4 1 7 4 2 3 2 1 0 1 1 3 3 3 0 0 3 4 2 5 4 4 1 3 3 0 4 2 0 4 1 0
 1992 1 1 3 0 2 1 -1 75 0 0 7 4 5 10 4 3 0 3 1 0 2 0 2 1 1 0 0 5 1 3 8 3 3 1 2 0 1 3 0 1 1 0
 1993 1 1 3 0 2 1 -1 75 0 0 7 4 3 7 5 7 2 1 0 1 0 4 0 0 0 0 0 3 3 4 3 7 0 0 4 2 1 1 1 5 0 0
 1994 1 1 3 0 2 1 -1 75 0 0 3 6 4 4 4 9 4 5 1 0 0 0 0 0 3 0 0 0 9 0 7 2 2 3 4 0 3 2 0 0 0 0
 1995 1 1 3 0 2 1 -1 75 3 1 2 0 8 5 2 6 2 5 0 2 1 4 0 0 0 0 0 0 2 5 3 2 3 5 6 1 0 1 1 3 1 1
 1996 1 1 3 0 2 1 -1 75 0 0 1 1 5 4 3 7 2 3 2 3 3 1 5 1 0 0 2 5 0 5 4 1 2 3 4 2 3 0 1 2 0 0
 1997 1 1 3 0 2 1 -1 75 0 5 3 5 0 2 4 3 4 5 1 1 3 2 2 0 0 0 0 0 3 1 6 5 5 2 3 4 1 2 3 0 0 0
 1998 1 1 3 0 2 1 -1 75 5 3 1 4 1 2 3 4 3 2 0 2 0 1 5 0 0 0 0 4 6 4 2 7 2 1 1 6 3 0 0 2 1 0
 1999 1 1 3 0 2 1 -1 75 2 2 3 3 6 3 3 3 8 3 3 3 0 1 1 0 0 0 1 3 3 3 5 4 0 4 2 4 0 1 0 1 0 0
 2000 1 1 3 0 2 1 -1 75 0 2 1 9 4 4 2 2 4 3 1 0 1 0 5 0 0 0 0 8 11 3 1 2 2 1 1 2 1 0 2 3 0 0
 2001 1 1 3 0 2 1 -1 75 0 1 1 6 8 1 1 0 5 2 2 2 0 3 4 0 0 0 0 5 3 4 6 3 3 1 4 3 1 1 2 3 0 0
 1977 1 2 3 0 2 1 -1 75 2 1 2 1 0 4 3 3 2 1 1 0 1 1 4 7 0 0 2 2 7 1 0 1 0 1 2 4 1 2 2 7 10 0
 1980 1 2 3 0 2 1 -1 75 3 3 4 6 5 2 0 2 3 0 3 2 2 2 2 1 4 0 2 3 5 3 1 2 1 1 2 1 1 1 0 3 1 4
 1983 1 2 3 0 2 1 -1 75 3 4 3 2 3 0 0 7 0 0 3 1 1 0 5 6 0 0 2 2 4 1 2 3 4 3 2 0 1 1 2 7 1 2
 1986 1 2 3 0 2 1 -1 75 3 0 2 5 3 5 5 3 1 3 2 1 1 1 3 0 2 0 0 2 3 6 6 1 3 3 1 1 1 1 2 2 3 0
 1989 1 2 3 0 2 1 -1 75 7 3 7 3 2 1 0 3 2 1 2 1 1 5 0 0 0 0 4 8 6 1 2 3 5 1 1 2 0 4 0 0 0 0
 1992 1 2 3 0 2 1 -1 75 2 5 3 4 0 5 0 5 2 0 0 0 1 0 3 0 0 0 4 5 5 10 8 6 2 1 2 0 0 1 0 1 0 0
 1995 1 2 3 0 2 1 -1 75 0 5 2 3 2 3 5 4 2 1 1 2 0 0 3 0 0 0 2 3 5 11 2 6 5 1 2 1 2 0 0 2 0 0
 1998 1 2 3 0 2 1 -1 75 9 4 4 3 1 1 1 1 3 3 1 2 1 7 0 0 0 0 6 5 3 5 1 3 3 2 3 2 0 1 0 0 0 0
 2001 1 2 3 0 2 1 -1 75 4 0 4 11 5 3 4 2 2 0 0 0 0 0 2 0 0 0 2 4 7 11 5 2 0 2 2 2 0 0 0 1 0 0
```

```
#
4 #_N_MeanSize-at-Age_obs
#Yr Seas Flt/Svy Gender Part Ageerr Ignore datavector(female-male)
#                                          samplesize(female-male)
 1971 1 1 3 0 1 2 29.8931 40.6872 44.7411 50.027 52.5794 56.1489 57.1033 61.1728 61.7417 63.368 64.4088 65.6889 67.616
68.5972 69.9177 71.0443 72.3609 32.8188 39.5964 43.988 50.1693 53.1729 54.9822 55.3463 60.3509 60.7439 62.3432 64.3224
65.1032 64.1965 66.7452 67.5154 70.8749 71.2768 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20
 1995 1 1 3 0 1 2 32.8974 38.2709 43.8878 49.2745 53.5343 55.1978 57.4389 62.0368 62.1445 62.9579 65.0857 65.6433
66.082 65.6117 67.0784 69.3493 72.2966 32.6552 40.5546 44.6292 50.4063 52.0796 56.1529 56.9004 60.218 61.5894 63.6613
64.0222 63.4926 65.8115 69.5357 68.2448 66.881 71.5122 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20
 1971 1 2 3 0 1 2 34.1574 38.8017 43.122 47.2042 49.0502 51.6446 56.3201 56.3038 60.5509 60.2537 59.8042 62.9309 66.842
67.8089 71.1612 70.7693 74.5593 35.3811 40.7375 44.5192 47.6261 52.5298 53.5552 54.9851 58.9231 58.9932 61.8625 64.0366
62.7507 63.9754 64.5102 66.9779 67.7361 69.1298 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20
 1995 1 2 3 0 1 2 34.6022 38.3176 42.9052 48.2752 50.6189 53.476 56.7806 59.4127 60.5964 60.5537 65.3608 64.7263
67.4315 67.1405 68.9908 71.9886 74.1594 35.169 40.2404 43.8878 47.3519 49.9906 52.2207 54.9035 58.6058 60.0957 62.4046
62.2298 62.1437 66.2116 65.7657 69.9544 70.6518 71.4371 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20
#
0 #_N_environ_variables
0 #_N_environ_obs
0 # N sizefreq methods to read
#
0 # no tag data
#
0 # no morphcomp data
#
999
#
```

References

Chen, Y., Jiao, Y., Chen, L. 2003. Developing robust frequentist and Bayesian assement methods. Fish and Fisheries. 4: 105-120.

Fournier, D.A., H.J. Skaug, J. Ancheta, J. Ianelli, A. Magnusson, M.N. Maunder, A. Nielsen, and J. Sibert. 2012. AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. Optim. Methods Softw. 27:233-249.

Maunder, M.N., and R.B. Deriso. 2003. Estimation of recruitment in catch-at-age models. Can. J. Fish. Aquat. Sci. 22:1204-1216.

Methot, R.D., and I.G. Taylor. 2011. Adjusting for bias due to variability of estimated recruitments in fishery assessment models. Can. J. Fish. Aquat. Sci. 68: 1744-1760.

Methot, R.D., and C.R. Wetzel. 2013. Stock Synthesis: A biological and statistical framework for fish stock assessment and fishery management. Fish. Res. 142: 86-99.

Taylor, I.G., Gertseva, V., Methot., R.D., and M.N. Maunder. 2013. A stock recruitment relationship based on pre-recruit survival, illustrated with application to spiny dogfish shark. Fish. Res. 142: 15-21.