

User Manual Stock Synthesis

Version 3.30 beta

Jan 30, 2016

Richard D. Methot
Teresa A'mar
Chantel Wetzel

NOAA Fisheries
Seattle, WA USA

Contents

List of Figures	i
1 Introduction	1
2 New Features 3.30	1
3 File Organization	1
3.1 Input Files	1
3.2 Output Files	2
3.3 Auxiliary Files	3
4 Starting SS	4
5 Computer Requirements and Recommendations	4
6 Starter	4
6.1 Jitter	12
7 Forecast	12
8 Optional Inputs	18
8.1 Empirical Weight-at-Age (wtatage.ss)	18
8.2 runnumbers.ss	19
8.3 profilevalues.ss	20
9 Data File	20
9.1 Overview of Data File	20
9.2 Units of Measure	21
9.3 Time Units	21
9.4 Data File Syntax	22
9.4.1 Model Dimensions	22
9.4.2 Fleet Definitions	22
9.4.3 Catch	24
9.4.4 Abundance Indices	25
9.4.5 Discard	27
9.4.6 Mean Body Weight	29
9.4.7 Population Length Bins	30
9.4.8 Length Composition Data	32
9.4.9 Age Composition Bin Setup	33
9.4.10 Ageing Error	34
9.4.11 Conditional Age'-at-Length	36
9.4.12 Sex Ratio-at-Length	36

9.4.13	Mean Length or Body Weight-at-Age	37
9.4.14	Environmental Data	38
9.4.15	Generalized Size Composition Data	39
9.4.16	Tag-Recapture Data	40
9.4.17	Stock Composition Data	42
9.4.18	Excluding Data	43
9.4.19	Data Super Periods	43
10	Control File	44
10.1	Overview of Control File	44
10.2	Parameter Line Elements	46
10.3	Control File Syntax	47
10.3.1	Biology	49
10.3.2	Read Mortality-Growth Parameters	53
10.3.3	Natural Mortality Notes	57
10.3.4	Growth Notes	57
10.3.5	Growth Patterns (morphs) and Sub-Morphs	58
10.3.6	Recruitment, Age, and Growth	58
10.3.7	Cohort Growth Deviation	59
10.3.8	Movement Parameters	59
10.3.9	Recruitment Allocation and Movement Parameters	60
10.3.10	Catch Multiplier	62
10.3.11	Ageing Error Parameters	62
10.3.12	Time-Varying Biology Parameters	63
10.3.13	Notes on Seasonal Biology Parameters	65
10.3.14	Empirical Weight-at-Age (wtatage.ss)	65
10.3.15	Spawner-Recruitment	65
10.3.16	Spawner-Recruitment Function	70
10.3.17	Recruitment Eras	74
10.3.18	Recruitment Likelihood	74
10.3.19	Recruitment Bias Adjustment	74
10.3.20	Recruitment Autocorrelation	76
10.3.21	Recruitment Cycle	76
10.3.22	Initial Age Composition	77
10.3.23	Fishing Mortality Method	77
10.3.24	Initial Fishing Mortality	78
10.3.25	Catchability	79
10.3.26	Selectivity and Discard	82
10.3.27	Selectivity Patterns	83
10.3.28	Selectivity Pattern Details	85
10.3.29	Retention	95
10.3.30	Discard Mortality	96
10.3.31	Male Selectivity	96

10.3.32	Reading the Selectivity and Retention Parameters	98
10.3.33	Tag Recapture Parameters	99
10.3.34	Variance Adjustment Factors	100
10.3.35	Lambdas (Emphasis Factors)	101
10.3.36	Controls for Variance of Derived Quantities	103
10.3.37	Time-Varying Parameter Options	105
10.3.38	Parameter Priors	108
11	Output Files	112
11.1	Standard ADMB output files	112
11.2	Derived Quantities	112
11.2.1	Metric for Fishing Mortality	112
11.2.2	Equilibrium SPR	112
11.2.3	F std	113
11.2.4	F-at-Age	114
11.2.5	MSY and other Benchmark Items	114
11.3	Brief cumulative output	115
11.4	Output for Rebuilder Package	115
11.5	Bootstrap Data Files	118
11.6	Forecast and Reference Points	118
11.7	Main Output File, report.sso	123
12	Running SS	128
12.1	Example of DOS batch input file	129
12.1.1	Simple Batch	129
12.1.2	Complicated Batch	130
12.1.3	Batch Using PROFILEVALUES.SS	130
12.1.4	Re-Starting a Run	131
12.1.5	Debugging Tips	132
12.1.6	Keyboard Tips	132
12.1.7	Running MCMC	132
12.1.8	Using R To View Model Output	134

List of Figures

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 or GitHub. Additional information about the package can be located at <https://github.com/r4ss/r4ss>.

2 New Features 3.30

Stock Synthesis version 3.30 has a number of new features:

1. Version conditional read
2. Forecast allocation group by year
3. Subseasons
4. Fleets - flexible ordering
5. Catch - revised input format
6. Catch multiplier
7. Catch - bycatch fleets

3 File Organization

3.1 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)

3.2 Output Files

1. ss3.par, ss3.std, ss33.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).

3.3 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

4 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.

5 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.

6 Starter

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 the line is free form	All lines in this file beginning with #C will be retained and written to the top of several output files
data_file.dat	data_file.dat	File name of the data file
control_file.ctl	control_file.ctl	File name 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-structure report 0 = omit catch-at-age for each fleet and cohort 1 = include all output	Detailed age-structured report in REPORT.SSO.
0	Check-up 0 = omit 1 = write detailed intermediate calculations to ECHOINPUT.SSO during first call	This output is largely unformatted and undocumented and is mostly used by the developer.

Typical Value	Options	Description
0	Parameter Trace 0 = omit 1 = write good iteration and active parms 2 = write good iterations and all parms 3 = write every iteration and all parms 4 = write every iteration 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.

∞

Typical Value	Options	Description
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 added error 3+ = 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 is multiplied by a random normal deviation $rdev = N(0,1)$ to a transformed parameter value based upon the predefined parameter bounds. <i>click here for more information</i>
-1	SD Report Start -1 = begin annual SD report in start year <year> = begin SD report this year	

Typical Value	Options	Description
-1	SD Report End -1 = end annual SD report in end year -2 = end annual SD report in last forecast year <value> = end SD 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.
0	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 = $X*B_0$ 2 = $X*B_{MSY}$ 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 of $SSBy/(0.40*SSB_0)$

Typical Value	Options	Description
1	SPR report basis 0 = skip 1 = use $1 - \text{SPR}_{\text{TARGET}}$ 2 = use $1 - \text{SPR}$ at MSY 3 = use $1 - \text{SPR}$ at B_{TARGET} 4 = no denominator, so report actual $1 - \text{SPR}$ 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 $(1.0 - \text{SPR})$. 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 as $\ln(N(t+1)/N(t))$ 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.

Typical Value	Options	Description
COND: If F std reporting = 4	13 17 Age range if F std 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	999: Indicates that the control and data file are in a previous SS 3.24 version and will be converted to the new formatting in the control.ss_new and data.ss_new files. 3.3: Indicates that the control and data files are currently in SS3.30 format. 3.3: Indicates that the control and data files are currently in SS3.30 format.	SSv3.30 will create converted files in the new format from previous version when 999 is given. All ss_new files are in the 3.30 format, so starter.ss_new has 3.30 on the last line. Some Mgpargs are in new sequence, so 3.30 cannot read a ss3.par file produced by version 3.24 and earlier.
End of Starter File		

6.1 Jitter

The jitter factor is multiplied by a random normal deviation $rdev = N(0,1)$ to a transformed parameter value based upon the predefined parameter bounds:

$$temp = -\frac{1}{2}rdev \times jitter \times \ln\left(\frac{P_{MAX} - P_{MIN} + 0.0000002}{P_{VAL} - P_{MIN} + 0.0000001} - 1\right) \quad (1)$$

with the final jittered starting parameter value backtransformed as:

$$P_{NEW} = P_{MIN} + \frac{P_{MAX} - P_{MIN}}{1 + e^{-2 \times temp}} \quad (2)$$

7 Forecast

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
1	Benchmarks/Reference Points 0 = omit 1 = calculate 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 1 = F(SPR) 2 = F(MSY) 3 = F(Btarget) 4 = F(end year) 5 = Average recent F(enter years) - 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 average F relative to the end year. 0 Last year for recent average F.	Read a range of years for calculation of recent average F (not yet implemented). 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.

Typical Value	Options	Description
0.40	Relative 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 beg. bio; end bio; beg. selex; end selex; beg relF; end relF; >0 = absolute year <= 0 = year relative to end year	Requires 6 values, beginning and ending years for biology, selectivity, and relative Fs, that will be used in to calculate benchmark quantities
1	Benchmark Relative F Basis 1 = use year range 2 = set range for relF same as forecast below	Does not affect year range for selectivity and biology.
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 average F scalar for the forecast relative F years below 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 F_{BTGT} 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 (must be ≥ 1)	At least one forecast year now required which differs from version 3.24 that allowed zero forecast years.
1	F scalar	Only used if Forecast option = 5 (input annual F scalar).

Typical Value	Options	Description
0 0 0 0	Forecast Years Begin selex; end selex; begin relative F; end relative F >0 = absolute year <= 0 = year relative to end year	Requires 4 values: 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.
1	Control Rule 1 = catch = F(SSB) U.S. West Coast 2 = F = F(SSB)	
0.40	Control Rule Upper Limit	Biomass level (as a fraction of SB0) above which F is constant.
0.10	Control Rule Lower Limit	Biomass level (as a fraction of SB0) above which F is set to 0.
0.75	Control Rule Buffer	Multiplier applied to forecast F before calculating catch.
3	Number of forecast loops (1,2,3)	Maximum number of forecast loops: 1=OFL only, 2=ABC control rule, 3=set catches equal to control rule or input catch and redo forecast implementation error.
3	First forecast loop with stochastic recruitment	If this is set to 1 or 2, then OFL and ABC will be as if there was perfect knowledge about recruitment deviations in the future.
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	Implementation Error	The standard deviation of the log of the ratio between the realized catch and the target catch in the forecast. (set value
0	Rebuilder 0 = omit West Coast rebuilder output 1 = do rebuilder output	> 0.0 to cause active implementation error).

Typical Value	Options	Description
2004	Rebuilder catch (Year Declared) >0 = year first catch should be set to zero -1 = set to 1999	
2004	Rebuilder start year (Year Initial) >0 = year for current age structure -1 = set to end year +1	
1	Fleet Relative F 1 = use first-last allocation year 2 = read season(row) x fleet (column) set below	
2	Basis for maximum forecast catch 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 total numbers	
COND: 2 (Conditional input for fleet relative F)		
0.1 0.8 0.1	Fleet allocation by relative 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. Starting in version 3.3 this now also includes surveys which are treated similar to fleets. Ex: # Fleet 1 Fleet 2 Survey X 0.10 0.10 0.30 # Area1 0.10 0.10 0.30 # Area2

Typical Value	Options	Description
1 50 -9999 -1	Maximum total catch by fleet -1 = no maximum	Must enter value for each fleet. Starting in version 3.3 this now also includes surveys which are treated similar to fleets.
-9999 -1	Maximum total catch by area -1 = no maximum	Must enter value for each area. Starting in version 3.3 this now also includes surveys which are treated similar to fleets.
1 1 -9999 -1	Fleet assignment to allocation group 0 = Fleet not included in allocation group	Enter group ID # for each fleet. Starting in version 3.3 this now also includes surveys which are treated similar to fleets.
COND: >0		
2002 1 -9999 1	Allocation to each group for each year of the forecast	For each year of the forecast, enter the allocation fraction to each group. Annual values are rescaled to sum to 1.0.
-1	Basis for forecast catch -1 = Read basis with each observation, allows for a mixture of dead, retained, or F basis by different fleets for the fixed catches below. 2 = Dead catch 3 = Retained catch 99 = Input harvest rate (F)	
COND: >0	Forecasted catches - enter one line per number of fixed forecast year catch	
2012 1 1 1200 2	Year Season Fleet Catch (of F value) Basis	
2013 1 1 1200 2	Year Season Fleet Catch (of F value) Basis	
-9999 1 1 0 2	Year Season Fleet Catch (of F value) Basis	
999	End of Input	
End of Forecast File		

8 Optional Inputs

8.1 Empirical Weight-at-Age (wtatage.ss)

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 equal to 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	# Number of rows								
40	# Number of ages (equal to Maximum Age)								
#Year	Season	Gender	GP	Birth Season	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.

8.2 runnumbers.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.

8.3 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.

9 Data File

9.1 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. Age imprecision definitions
10. Age composition
11. Mean length of bodyweight-at-age
12. Generalized size composition (e.g. weight frequency)
13. Tag-recapture
14. Stock composition
15. Environmental data

9.2 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 $\text{eggs/kg} = 1$ for all weights; otherwise has units that are proportional to egg production

9.3 Time Units

Year

- Spawning is restricted to happening once per year at a specified time of year (in real months).
- Time-varying parameters are allowed to change annually.
- Rates like growth and mortality are per year.
- All fish advance to the next older integer age on January 1, no matter when they were born during the year

Seasons

- Seasons are the time step during which constant rates apply
- Seasons are the time step for which catch and discard is input and for which F is calculated
- The year can have just 1 annual season, or be subdivided into seasons of unequal length.
- Season duration is input in real months and is converted into fractions of an annum. Annual rate values are multiplied by the per annum season duration.
- If the sum of the input season durations is not close to 12.0, then the input durations is divided by 12. This allows for a special situation in which the year could be only 0.25 in duration (e.g. seasons as years) so that spawning and time-varying parameters can occur more frequently.

9.4 Data File Syntax

9.4.1 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	Number of 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 now calculated in v3.3 by season (growth, mortality, etc.).
2	The number of subseasons. Entry must be even and the minimum value is 2. This is for the purpose of finer temporal granularity in calculating and using the length-at-age.
1	Spawning subseason; spawning biomass is calculated at this time of year and used as basis for the total recruitment of all settlement events resulting from this spawning.
2	Number of genders
20	Number of ages. The value here will be the plus-group age. SS start age 0.
1	Number of areas
2	Number of fleets (including surveys)

9.4.2 Fleet Definitions

The catch data input has been modified to improve the user flexibility to add/subtract fishing and survey fleets to a model set-up. The fleet setup input is transposed so each fleet is now a row. Previous versions (3.24 and earlier) required that fishing fleets be listed first followed by survey only fleets. In version 3.30 all fleets now have the same status within the model structure and each has a specified fleet type. Available types are: catch fleet, bycatch only, or survey.

Inputs that define the fishing and survey fleets:

2	#Number of fleets which includes survey in any order						
#Fleet Type	Timing	Area	Catch Units	Eq. Catch SE	Catch SE	Catch Mult.	Fleet Name
1	-1	1	1	0.01	0.01	0	FISHERY1
1	-1	1	2	0.01	0.01	0	SURVEY1

Fleet Type

- 1 = fleet with input retained catch
- 2 = bycatch fleet
- 3 = survey
- 4 = ignored (not yet implemented)

Timing

Carryover from 3.24 approach, now superseded by real month input for data observations.

- 0.50 = now ignored in v3.30
- -1 = treat as catch from whole season

Area

An integer value indicating the area in which a fleet operates.

Catch Units

Ignored for survey fleets, their units are read later

- 1 = biomass
- 2 = numbers

Equil. Catch SE

Standard error of the initial equilibrium catch.

Catch SE

Standard error of retained catch; ignored for survey fleets and bycatch fleets and with Pope's F method.

Catch Multiplier

Invokes use of a catch multiplier, which is then entered as a parameter in the MG parameter section. The estimated value or fixed value of the catch multiplier is multiplied by the estimated catch before being compared to the observed catch.

- 0 = no catch multiplier used

- 1 = Apply a catch multiplier which is defined as an estimatable parameter in the control file after the cohort growth deviation in the biology parameter section. The model's estimated retained catch will be multiplied by this factor before being compared to the observed retained catch.

After reading the fleet-specific indicators, a list of catch values by fleet and season are read in by the model. The format for the catches is year, season that the catch will be attributed to, fleet, a catch value, and a year specific catch standard error. To include an equilibrium catch value the year should be noted as -999 and be included as the first entry for the associated fleet. There is no longer a need to specify the number of catch records to be read; instead the list is terminated by entering a record with the value of -9999 in the year field. Additionally, initial equilibrium catch is now season specific and can be specified using -999 in the year column.

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.

If a bycatch fleet is included the continuous F method must be selected ($F_method = 2$) and are excluded from the catch log-likelihood. Bycatch fleets have selectivity and retention functions, so even though they are considered to have unknown catch levels, this does not mean that their calculated retained catch is zero. MSY and yield per recruit are calculated in terms of dead catch, and they currently include catch from bycatch fleets. The F for bycatch only fleets is kept constant in benchmark and forecast calculations, so it is not included in any forecast cap and allocation calculations. It is not part of the acceptable biological catch, but it is still calculated and reported.

9.4.3 Catch

The new format for version 3.30 for a 2 season model with 2 fisheries looks like the table below. The example is sorted by fleet, but the sort order does not matter. In data.ss_new, the sort order is fleet, year, season.

#Catches by year, season for every fleet:

# Year	Season	Fleet	Catches	Catch SE
-999	1	1	56	0.05
-999	2	1	62	0.05
1975	1	1	876	0.05
1975	2	1	343	0.05
...
-999	1	2	55	0.05
-999	2	2	22	0.05
1975	1	2	555	0.05
1975	2	2	873	0.05
...
-9999	0	0	0	0

- Catch can be in terms of biomass or numbers for each fleet.
- Catch is retained catch. If there is discard 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. However, this is better handled with the new catch multiplier option.

If a bycatch fleet is used, continuous F (F_method 2) must be used and are excluded from the catch log likelihood. Bycatch fleets have selectivity and retention functions, so even though they are considered to have unknown catch levels, this does not mean that their retained catch is zero. SS v3.30 will later add the option for bycatch fleets to have retained and discarded catch calculated or have all their catch be assigned to discard. MSY and yield per recruit are calculated in terms of dead catch, and currently include catch from bycatch fleets. Future bycatch fleet options will address this.

9.4.4 Abundance Indices

For fishing fleets, catch-per-unit-effort (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.

#CPUE and Suvey Abundance Observations:

#Fleet/Survey	Units	#Error Distribution		
1	1	0		
2	1	0		
...		
#Year	Month	Fleet/Survey	Observation	SE of log(value)
1991	7	3	80000	0.056
1995	7.2	3	65000	0.056
...
2000	7.1	3	42000	0.056
-9999	1	1	1	1

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 $\log_e(\text{index})$. If the variance of the observations is available only as a CV, then the value of se can be approximated as $\sqrt{(\log_e(1 + (CV)^2))}$ 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 $\exp(\text{Z-score})$ 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 $-\log L$ 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.

- 30 = spawning biomass (e.g. for an egg and larvae survey)
- 31 = $\exp(\text{recruitment deviation})$, useful for environmental index affecting recruitment
- 32 = spawning biomass * $\exp(\text{recruitment deviation})$, for a pre-recruit survey occurring before density-dependence
- 33 = recruitment, age-0 recruits

9.4.5 Discard

If discard is not a feature of the model specification, then just a single input is needed:

#Input	Description
0	#Number of fleets with discard observations

If discard is being used, the input syntax is:

#Input	Description			
1	#Number of fleets with discard observations			
#Fleet	Units	Error		
1	2	-1		
#Year	Month	Fleet	Observation	Error
1980	7	1	0.05	0.25
1991	7	1	0.10	0.25
-9999	1	1	0	0

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 discard error 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

- Since discard refers to catch, its time units are in seasons, not months.
- 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 log likelihood.

- Zero (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 standard error (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 standard error, not to the CV, to help correct this problem of underestimated variance.

9.4.6 Mean Body Weight

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 obtained 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.

#Mean Body Weight Data Section					
#Degrees of freedom for Student's t-distribution used to evaluate mean body weight deviation. This is not a conditional input, must be here even if there are no mean body weight observations.					
30					
#Year	Month	Fleet	Partition	Value	CV
1990	7	1	0	4.0	0.95
1990	7	1	0	1.0	0.95
-9999	1	1	0	0	0

Partition

Mean weight data and composition data require specification of what group the sample originated from (e.g. discard, retained, discard + retained).

- 0 = whole catch in units of weight (discard + retained)
- 1 = discarded catch in units of weight
- 2 = retained catch in units of weight
- 10 = whole catch in units of length (discard + retained)
- 11 = discarded catch in units of weight

- 12 retained catch in units of length

Value - Units

Units must correspond to the units of body weight (or mean length in cm), normally in kilograms. The expected value of mean body weight (or mean length) is calculated in a way that incorporates effect of selectivity and retention.

Error

Error is entered as the CV of the observed mean body weight (or mean length)

9.4.7 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.

1	1 = use data bins 2 = generate from bin width min max below 3 = read vector	Length bin method - creates a conditional read situation below.
COND = 1	Selects option 1, no additional input necessary	
COND = 2	Selects option 2, read 3 additional input values.	
	2	Bin width
	10	Lower size of first bin
	82	Lower size of largest bin
The number of bins is then calculated from: $(\max \text{Lread} - \min \text{Lread}) / (\text{bin width}) + 1$		
COND = 3	Selects option 3 - read 1 value and 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, bin width 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.
- 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 log likelihood with respect to that parameter and convergence will be hampered. A generic guidance to avoid this situation is not yet available.

Specify bin compression and error structure for length composition data:

#Min Tail	Constant	Combine	Compress	Error	Dirichlet	
Compression	added to	males &	Bins	Distribution	Parameter	
	proportions	females			Number	
0	0.0001	0	0	0	0	#Fleet 1
0	0.0001	0	0	0	0	#Fleet 2

Minimum Tail Compression

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.

Added Constant

Constant added to observed and expected proportions at length and age to make logL calculations more robust. Tail compression occurs before adding this constant. Proportions are renormalized to sum to 1.0 after constant is added.

Combine Males & Females

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.

Error Distribution

- 0 = Multinomial Error
- 1 = Dirichlet Error

Dirichlet Parameter Number

If the dirichlet error distribution is selected a number of parameters must be specified.

Notes:

- 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.
- 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.

9.4.8 Length Composition Data

30	#Number of length bins for data
26 28 30 ... 88 90	#Vector of length bins associated with the length data

Example of a single length composition observation:

#Year	Month	Fleet	Gender	Partition	Nsamp	data vector
1986	1	1	3	0	20	<female then male data>
...
-9999	0	0	0	0	0	0

Gender

If model has only one gender defined in the set-up, all observations must have gender set equal to 0 or 1. 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.

- 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

Partition indicates samples from either discards, retained, or combined.

- 0 = combined
- 1 = discard
- 2 = retained

Excluding Data

- 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.

Note:

- Version 3.30 no longer requires that the number of length composition lines to be read be specified. Entering -9999 at the end of the data matrix will indicate to the model the end of length composition lines to be read.
- 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.

9.4.9 Age Composition Bin Setup

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'.

17	#Number of 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.															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	20	25

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.

9.4.10 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			
# Age-0	Age-1	Age-2	...	Max Age
-1	-1	-1	...	-1
0.001	0.001	0.001	...	0.001
0.5	1.5	2.3	...	Max Age + 0.5
0.5	0.65	0.67	...	4.3

The above table shows the values for the first 3 ages for each of two age transition definitions: the first defines a matrix with no bias and negligible imprecision and the second shows a small negative bias beginning at age 2.

Note:

- If no age data, there can be 0 vectors.
- In principle, one could have year or laboratory specific matrices.
- 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. The following line is a a vector with the standard deviation (stddev) of age' for each true age.
- SS is able to create one ageing error matrix from parameters, rather than from an input vector. The range of conditions in which this new feature 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 age error 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.

Specify bin compression and error structure for age composition data:

#Min Tail	Constant	Combine	Compress	Error	Dirichlet	
Compression	added to proportions	males & females	Bins	Distribution	Parameter Number	
0	0.0001	1	0	0	0	#Fleet 1
0	0.0001	1	0	0	0	#Fleet 2
0	0.0001	1	0	0	0	#Survey 2
1	Bin method for age data 1 = value refers to population bin index 2 = value refers to data bin index 3 = value is actual length (which must correspond to population length bin boundary)					

An example age composition observation:

#Year	Month	Fleet	Gender	Partition	AgeErr	Lbin lo	Lbin hi	Nsamp	Data Vector
1987	1	1	3	0	2	-1	-1	79	<enter data values>
-9999	0	0	0	0	0	0	0	0	0

Note:

- 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 fleet 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.

9.4.11 Conditional Age'-at-Length

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 for several reasons. First, it avoids double use of fish for both age' and size information because the age' information is considered conditional on the length information. Second, it 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. Lastly, 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.

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.

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.

9.4.12 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.

#Example setup for sex ratio-at-length data:

1	#N age bins so all fish are put into a single "age" bind regardless of their true age														
10	#Assigned "age" for this one bin														
1	#N of age error 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	# Lbin method: 1 = population length bins, 2 = data length bins, 3 = lengths														
0	#Combine males and females at or below this bin number														
#There are 4 females and 8 males in the 25th population length bin															
#Yr	Month	Fleet	Gender	Part	AgeErr	Lbinlo	Lbinhi	Nsamp							
1971	1	1	3	0	1	25	25	12	0	4	0	0	8	0	...
-9999	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 number of 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 so all fish are put into a single "age" bind regardless of their true age												
1 2 3	#Assigned "age" for this one bin												
4 5													
2	#N of age error definitions												
-1	1	1	1	...	repeat for each true age in model, beginning at age-0								
0.2	0.4	0.5	0.8	...	repeat for each true age in model, beginning at age-0								
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								
1	# Lbin method: 1 = population length bins, 2 = data length bins, 3 = lengths												
0	#Combine males and females at or below this bin number												
#There are 4 females and 8 males in the 25th population length bin													
#Yr	Month	Fleet	Gender	Part	AgeErr	Lbinlo	Lbinhi	Nsamp					
1971	1	1	3	0	1	-1	-1	25	1	2	4	...	#real age data 5
1971	1	1	3	0	1	25	25	12	0	0	4	...	#sex ratio in bin 3
-9999	1	1	3	0	1	25	25	12	0	0	4	...	

9.4.13 Mean Length or Body Weight-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:

#Yr	Month	Fleet	Gender	Part	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>
-9999	1	1	3	0	2	999				

Note:

- Nsamp value is ignored if positive, but a negative value will cause the entire observation to be ignore.
- Negatively valued mean size entries with 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 genders’ 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.

9.4.14 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	#Number of environmental variables	
# Example of 2 environmental observations:		
#Year	Variable	Value
1990	1	0.10
1991	1	0.15
-9999	0	0

Note:

- 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.

9.4.15 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 units of biomass or numbers.
- 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.

Example entry:

2		#N of weight frequency 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	#Min compression to add to each observation (entry for each method)
40	5	#N observations per weight frequency 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	2.4	4	9	1

Note:

- There is no tail compression 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	Year	Month	Fleet	Gender	Part	Sample Size	<composition females then males>
1	1975	1	1	3	0	43	<data>

9.4.16 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).

Example set-up for tagging data:

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	#Number of tag groups							
12	#Number of recapture events							
2	#Mixing latency period: N periods to delay before comparing observed to expected recoveries (0 = release period)							
10	#Max periods (months) to track recoveries, after which tags enter accumulator							
#Release Data								
#TG	Area	Year	Month	<tfill>	Gender	Age	N Release	
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		Month		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 enter in TG order.

- <tfill> values are place holders and are replaced by program generated values for model time.

9.4.17 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	#Number of observations							
2	#Number of stocks							
0.0001	#Minimum Compression							
#Year	Month	Fleet	Part	Nsamp	Data	Vector		
1980	1	1	0	36	0.4	0.6	...	
1981	1	1	0	40	0.44	0.62	...	
1982	1	1	0	50	0.49	0.50	...	

Note:

- 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 File

999	#End of data file marker
-----	--------------------------

9.4.18 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 negative log likelihood, are flagged by use of a negative value for fleet ID.

9.4.19 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 an 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 SS version 3.23 and higher. Previously, super-periods were started with a -9999 flag in a standard error (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 model version 3.23 and higher, super-periods are started with a negative value for season, and then stopped with a negative value for season, placeholder observations within the super-period are designated with a negative fleet field. The standard error (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 log likelihood calculation. An error message is generated if the super-period 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 combined data from all the specified time steps).

Example:

#Year	Month	Fleet	Obs	SE	Comment
1982	-2	3	34.2	0.3	Start super-period. This observation has positive fleet value, so is expected to contain combined data from all identified periods of the super-period. 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	0.3	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	0.40	Note that 1984 is not included in the supe-rperiod. Relative weight for 1985 is 0.4
1986	-2	-3	88	0.40	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 super-period observations. For composition data (which allow for replicate observations), a particular time steps observations could be entered as a member of a super-period 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.

10 Control File

10.1 Overview of Control File

These listed model features are denoted in the control file in the following order:

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. Specification for 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. Catch Multiplier
11. Fraction female
12. Environmental link parameters for any mortality-growth (MG) parameters that use a link
13. Time-varying setup for any MG parameters that use blocks
14. Seasonal effects on biology parameters
15. Phase for any MG parameters that use annual deviations
16. Spawner-recruitment parameters
17. Recruitment deviations
18. Method for calculating fishing mortality (F)
19. Initial equilibrium F for each fleet
20. Catchability (Q) setup for each fleet and survey
21. Catchability parameters
22. Length selectivity, retention, discard mortality setup for each fleet and survey
23. Age selectivity setup for each fleet and survey
24. Parameters for length selectivity, retention, discard mortality for each fleet and survey
25. Parameters for age selectivity for each fleet and survey
26. Environmental link parameters for any selectivity/retention parameters that use a link
27. Time-varying setup for any selectivity/retention parameters that use blocks
28. Phase for any selectivity/retention parameters that use random annual deviation

- 29. Tag-recapture parameters
- 30. Variance adjustments
- 31. Lambdas for likelihood components

10.2 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 with bias adjustment, 5 = gamma
5	Prior StDev	Standard deviation for the prior, used to calculate likelihood of the current parameter value. This value is ignored if prior type is -1.
6	PHASE	Phase in which parameter begins to be estimated. A negative value causes the parameter to retain its INIT value (or value read from the 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 deviation vector
10	Dev min yr	Beginning year for the deviation vector
11	Dev max yr	Ending year for the deviation vector
12	Dev StDev	Standard deviation for elements in the deviation vector
13	USE-BLOCK	Set up blocks or parameter trends
14	BLOCK-TYPE	Function form for the block offset

10.3 Control 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 conditions.

- In most cases, the description in the Definition column is the same as the label output to the control.ss_new file.

Typical Value	Description and Options
#C comment	Comments beginning with #C at the top of the file will be retained and included in output
2	N growth patterns (GP)
3	Number of sub-morphs within a growth pattern. Permissible 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 to first value of vector to get a normal approximation: (0.15, 0.70, 0.15) for 3 sub-morphs, (0.031, 0.237, 0.464, 0.237, 0.031) for 5 sub-morphs.
1	Recruitment distribution method. Options: 1 = use the 3.24 or earlier setup, 2 = main effects for GP, settle timing, and area, 3 = each settle entity, and 4 = none when N GP*Nsettle=1
1	Number of recruitment settlement assignments. Options: 1 = global, 2 = by area
COND = 1	Only read if recruitment distribution method is set to 1 (3.24 and earlier version)
0	Year x Area x Settlement Event Interaction Requested
1 1 1	Recruitment assignment to GP, season, area (for each settlement event).
COND:	If there are multiple GP, season, and areas, specify the additional lines:
1 1 1	Recruitment assignment to GP1, season 1, area 1
2 1 2	Recruitment assignment to GP1, season 1, area 2
2 1 1	Recruitment assignment to GP2, season 1, area 1
2 2 2	Recruitment assignment to GP2, season 2, area 2
COND:	If N areas > 1:
0	Movement: Only read following movement section if N areas > 1
COND > 0	Following lines are conditional if movement is selected:
4	N movement definitions
0.60	First age that moves. Real, not integer age at the beginning of season. This control primarily used to keep new recruits from moving until after their first year.

Typical Value	Description and Options
1 1 1 2 4 10 1 1 2 1 4 10 1 2 1 2 4 10 1 2 2 1 4 10	The four requested movement definitions appear here. Each definition specifies: season, 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.
3	Number of block patterns. These patterns can be referred to in the parameter sections to create a separate parameter value for each block.
COND: 3 2 1	Following inputs are omitted if N Block patterns equals 0 Blocks per pattern
1975 1985 1986 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.

10.3.1 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.

Typical Value	Description and Options
1	Natural Mortality Option: 0 = A single parameter 1 = N breakpoints 2 = Lorenzen 3 = Read age specific M and do not do seasonal interpolation 4 = Read age specific and do seasonal interpolation, if appropriate

Typical Value	Description and Options
COND = 0	No additional natural mortality 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	Lorenzen Natural Mortality: read one additional value that is the reference age (integer) (<i>click here for more information</i>). 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 a 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 2, 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.
0.20 0.25 ... 0.20 0.23 ...	Age-specific M values: row 1 is female GP1, row 2 is female 2 GP2, row 3 is male GP1, etc.

Typical Value	Description and Options
1	<p>Growth Model:</p> <p>1 = von Bertalanffy (2 parameters)</p> <p>2 = Schnute's generalized growth curve (aka Richards curve) with 3 parameters</p> <p>3 = von Bertalanffy with age-specific k deviations for specified range of ages</p>
1.66	Growth Amin (A1): Reference age for first size-at-age parameter (<i>click here for more information</i>)
25	Growth Amax (A2): Reference age for second size-at-age parameter.
COND = 3	Growth option age-specific k
5	Minimum age for age-specific k
7	Maximum age for age-specific k
0	Standard deviation added to length-at-age: Enter 0.10 to mimic SS2 V1.xx. Recommend using a value of 0.0. (<i>click here for more information</i>)
1	<p>CV Pattern</p> <p>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.</p> <p>1: $CV=f(A)$, so interpolation is a function of age.</p> <p>2: $SD=f(LAA)$, so parameters define the standard deviations of length-at-age and interpolation is a function of mean length-at-age.</p> <p>3: $SD=f(A)$</p> <p>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.</p>
2	<p>Maturity Option:</p> <p>1 = length logistic,</p> <p>2 = age logistic,</p> <p>3 = read age-maturity for each female GP</p> <p>4 = read an empirical age-maturity vector for all ages</p> <p>5 = read empirical age-fecundity and body weight-at-age from separate file, wtatage.ss. Allows for reading time series of input. See section <i>Empirical Wt-at-Age</i> for details. NOTE: need to read 2 parameters even if option 3, 4, or 5 is selected.</p>

Typical Value	Description and Options
	6 = read an empirical length-maturity vector by population length bin (available in v3.24q)
COND = 3 or 4	Maturity Option
0 0.05 0.10	Vector of age-specific maturity or fecundity. One row of length Nages + 1 for each female GP
...	
COND = 6	Maturity Option
0 0.05 0.10	Vector of length-specific maturity or fecundity. One row of length of the population length bins for each female GP
...	
1	First Mature Age: Overridden if maturity option is 3, 4, or 5, but still must exist here. Otherwise, all ages below the first mature age will have maturity set to zero.
1	<p>Fecundity Option (irrelevant if maturity option is 4 or 5):</p> <p>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 equivalent to spawning biomass.</p> <p>2 = to set fecundity = $a * L^b$</p> <p>3 = to set fecundity = $a * W^b$, so values of a=1, b=1 causes fecundity to be equiv to spawning biomass</p> <p>4 = fecundity = $a + b * L$</p> <p>5 = Eggs = $a + b * wt$</p>
0	<p>Hermaphroditism Option:</p> <p>0 = not used,</p> <p>1 = invoke female to male transition,</p> <p>-1 = invoke male to female transition</p>
COND = 1	Read 2 lines below if hermaphroditism. is selected; also read 3 parameters after reading the male weight-length parameter
-1	<p>Hermaphroditism Season:</p> <p>-1 to do transition at the end of each season (after mortality and before movement)</p> <p><positive integer> to select just one season</p>
1	<p>Include males in spawning biomass</p> <p>0 = no males in spawning biomass</p> <p>1 = simple addition of males to females</p> <p>xx = more options to come later</p>
2	<p>Parameter Offset Method</p> <p>1 = direct assignment</p>

Typical Value	Description and Options
	<p>2 = for each GP x gender, parameter defines offset from gender 1, offsets are in exponential terms, so for example, $\text{old_male_M} = \text{old_female_M} * \exp(\text{old_male_parameter})$</p> <p>3 = for each GP x gender, parameter defines offset from GP 1 gender 1. For females, given that “natM option” is breakpoint and there are two breakpoints, parameter defines offset from early age (e.g., $\text{old_female_M} = \text{young_female_M} * \exp(\text{old_female_M_parameter})$). For males, given that “natM option” is breakpoint and there are two breakpoints, parameter is defined as offset from females AND from early age (e.g., $\text{old_male_M} = \text{young_female_M} * \exp(\text{young_male_M_parameter}) * \exp(\text{old_male_M_parameter})$).</p>
1	<p>Time-varying adjustment constraint:</p> <p>1 = standard parameter adjustments for environmental, block, and deviations are not constrained by bounds</p> <p>2 = logistic transform - parameter adjustments use a logistic transformation to assure that adjusted parameter value stays within bounds of base parameter</p> <p>3: standard with no bound check</p>

10.3.2 Read Mortality-Growth Parameters

Next, SS reads the MG parameters in generally the following order (may vary based on selected options):

Parameter	Description
Females	Female natural mortality and growth parameters in the following order by GP
natM	Natural mortality for female GP1, where the number of natural mortality parameters depends on the option selected.
Lmin	Length at Amin (units in cm) for female, GP1
Lmax	Length at Amax (units in cm) for female, GP1
VBK	Von Bertalanffy growth coefficient (units are per year) for females, GP1

COND if growth type =2

Parameter	Description
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 deviations for first age in rage	
K deviations for next age in rage	
...	
K deviations for last age in rage	
CV young	Variability for size at age \leq AFIX (units are fraction) for females, GP1. Note that CV cannot vary over time, so do not set up env-link or a dev vector. Also, units are either as CV or as standard deviation, depending on assigned value of CV pattern.
CV old	Variability for size at age \geq AFIX (units are fraction) for females, GP1. For intermediate ages, do a linear interpolation of CV on means size-at-age. Note that the units for CV will depend on the CV pattern and the value of MGparm as offset.
COND: GP > 1	Repeat female parameters in the above order for GP2
Males	Male natural mortality and growth parameters in the following order by GP
natM	Natural mortality for male GP1, where the number of natural mortality parameters depends on the option selected.
Lmin	Length at Amin (units in cm) for male, GP1
Lmax	Length at Amax (units in cm) for male, GP1
VBK	Von Bertalanffy growth coefficient (units are per year) for males, GP1
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 deviations for first age in rage	
K deviations for next age in rage	
...	
K deviations for last age in rage	

Parameter	Description
CV young	Variability for size at age \leq AFIX (units are fraction) for males, GP1. Note that CV cannot vary over time, so do not set up env-link or a dev vector. Also, units are either as CV or as standard deviation, depending on assigned value of CV pattern.
CV old	Variability for size at age \geq AFIX (units are fraction) for males, GP1. For intermediate ages, do a linear interpolation of CV on means size-at-age. Note that the units for CV will depend on the CV pattern and the value of MGparm as offset.
COND: GP > 1	Repeat male parameters in the above order for GP2
Females	Weight length relationship parameters, maturity and fecundity
WtLen scale	Coefficient to convert length in cm to weight in kg for females
WtLen exp	Exponent in to convert length to weight for females
Mat-50	Maturity logistic inflection (in cm or years). Where female maturity-at-length (or age) is a logistic function: $maturity = 1/(1 + \exp(slope * (size - at - age - inflection)))$
Mat-slope	Logistic slope (must have negative value)
Eggs-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.
Eggs-beta	
COND: GP > 1	Repeat female parameters in the above order for GP2
Males	Weight length relationship parameters
WtLen scale	Coefficient to convert length in cm to weight in kg for males
WtLen exp	Exponent in to convert length to weight for males
COND: GP > 1	Repeat male parameters in the above order for GP2
COND: Hermaphrodism	3 parameters define a normal distribution for the transition rate of females to males
Inflect Age	Hermaphrodite inflection age
StDev	Hermaphrodite standard deviation (in age)
Asmp Rate	Hermaphrodite asymptotic rate
Recr Dist GP	Recruitment apportionment by GP, if multiple GP, multiple entries required

Parameter	Description
Recr Dist Area	Recruitment apportionment by area, if multiple areas, multiple entries required
Recr Dist Seas	Recruitment apportionment by season, if multiple seasons, multiple entries required
COND: N patterns x N areas x N seasons	If recruitment distribution interaction = 1 (on) Note that the order of recruitment distribution parameters has areas then seasons for main effect, and seasons then areas for interactions.
Cohort growth deviation	If no deviations the INIT set equal to 1
2 x N selected movement pairs	Movement parameters
COND: Ageing Error Catch Multiplier	The following lines are only required when the associated features are turned on Turned on in the data file Turned on in the data file
Fraction female	Fraction female by GP, if multiple GP, multiple entries required

Example format for MG parameter section:

LO	HI	INIT	Prior Value	<other entries>	Block Type	Parameter Label
0	0.50	0.15	0.1	...	0	#NatM_p_1_Fem_GP_1
0	45	21	36	...	0	#L_at_Amin_Fem_GP_1
40	90	70	70	...	0	#L_at_Amax_Fem_GP_1
0	0.25	0.15	0.10	...	0	#VonBert_K_Fem_GP_1
0.10	0.25	0.15	0.20	...	0	#CV_young_Fem_GP_1
0.10	0.25	0.15	0.20	...	0	#CV_old_Fem_GP_1
-3	3	2e-6	0	...	0	#Wtlen_1_Fem
-3	4	3	3	...	0	#Wtlen_2_Fem
50	60	55	55	...	0	#Mat50%_Fem
-3	3	-0.2	-0.2	...	0	#Mat_slope_Fem
-5	5	0	0	...	0	#Eggs/kg_inter_Fem
-50	5	0	0	...	0	#Eggs/kg_slope_wt_Fem
0	0.50	0.15	0.1	...	0	#NatM_p_1_Mal_GP_1
0	45	21	36	...	0	#L_at_Amin_Mal_GP_1
40	90	70	70	...	0	#L_at_Amax_Mal_GP_1

LO	HI	INIT	Prior Value	<other entries>	Block Type	Parameter Label
0	0.25	0.15	0.10	...	0	#VonBert_K_Mal_GP_1
0.10	0.25	0.15	0.20	...	0	#CV_young_Mal_GP_1
0.10	0.25	0.15	0.20	...	0	#CV_old_Mal_GP_1
-3	3	2e-6	0	...	0	#Wtlen_1_Mal
-3	4	3	3	...	0	#Wtlen_2_Mal
0.001	0.999	0.5	0.5	...	0	#FracFemale_GP_1

10.3.3 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.

Lorenzen Natural Mortality

Lorenzen natural mortality is based on the concept that natural mortality varies over the life cycle of a fish, which is driven by physiological and ecological processes.

10.3.4 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 Linf. 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 von Bertalanffy K parameter. When this fourth parameter has a value of 1.0, it is equivalent to the standard von Bertalanffy growth curve. When this function was first introduced in SS, it required that A0 be set to 0.0.

Age-specific K

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 birth year. There is no offset for birth season timing effects, nor is there any seasonal interpolation of the age-varying K.

10.3.5 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.

10.3.6 Recruitment, Age, and Growth

Recruitment can occur in any season. In older versions of SS one value of spawning biomass was calculated annually at the beginning of one specified spawning season and this spawning biomass produces one annual total recruitment value and this annual recruitment was distributed among seasons, areas, and growth types according to other model parameters. SSv3.3 allows for the spawning biomass in a season to produce recruitment that may vary over the year based on the spawning biomass which associated with the area and growth types according to the model parameterization. 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 $\ln(\text{apportionment weight})$, 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.

10.3.7 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.

10.3.8 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 $\ln(\text{movement coefficient})$). 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).

A new feature added in v3.3 allows for a multi-season setup to have a growth pattern (GP) to have some fish recruit in different “birthseasons”. The movement parameters are now 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.

10.3.9 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:

LO	HI	INIT	Prior Value	<other entries>	Block Type	Parameter Label
-4	4	0	1	...	0	#RecrDist_GP_1
-4	4	0	1	...	0	#RecrDist_GP_2
-4	4	0	1	...	0	#RecrDist_Area_1
-4	4	-4	1	...	0	#RecrDist_Area_2
-4	4	0	1	...	0	#RecrDist_Seas_1
-4	4	-4	1	...	0	#RecrDist_Seas_2
-4	4	0	1	...	0	#CohortGrowthDev
COND: Only if movement is defined						
-5	5	-4	1	...	0	#MoveParm_A_seas_1_GP_1from_1to2
-5	5	-4	1	...	0	#MoveParm_B_seas_1_GP_1from_1to2
-5	5	-4	1	...	0	#MoveParm_A_seas_1_GP_1from_2to1
-5	5	-4	1	...	0	#MoveParm_B_seas_1_GP_1from_2to1
-5	5	-4	1	...	0	#MoveParm_A_seas_1_GP_2from_1to2
-5	5	-4	1	...	0	#MoveParm_B_seas_1_GP_2from_1to2
-5	5	-4	1	...	0	#MoveParm_A_seas_1_GP_2from_2to1
-5	5	-4	1	...	0	#MoveParm_B_seas_1_GP_2from_2to1

Note:

- 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:

$$rate_i = \frac{e^{p_i}}{\sum_{j=1}^N e^{p_i}} \quad (3)$$

- 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.

10.3.10 Catch Multiplier

This parameter line is only included in the control file if the catch multiplier field in the data file is set to 1. A single value may be fixed or estimated where:

$$C_{obs} = C_{exp} * c_{mult} \quad (4)$$

where C_{exp} is the expected catch and c_{mult} is the catch multiplier. It has year-specific, not season-specific, time-varying capabilities. In the catch likelihood calculation, expected catch is multiplied by the catch multiplier by year and fishery before being compared to the observed retained catch, so value of 1.1 means that the observed catch has overestimated actual catch by 10%.

10.3.11 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 is 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:

- age at which the estimated pattern begins (just linear below this age). This is “start age”
- bias at start age (as additive offset from unbiased age’)
- bias at maxage (as additive offset from unbiased age’)
- power fxn coefficient for interpolating between those 2 values (value of 0.0 produces linear interpolation in the bias)
- stdev at age
- stdev at max age
- power fxn coefficient for interpolating between those 2 values

10.3.12 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.

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 growth parameters in effect during the forecast will revert to the “no-block” baseline level. By setting the end year of the last block to end year (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 end year (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 L_{inf} 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.

Control file continued:

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 (<i>click here for more information</i>)

Value	Description
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.

10.3.13 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.

$$P' = P * \exp(seas_value) \quad (5)$$

10.3.14 Empirical Weight-at-Age (wtatage.ss)

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. See section 8.1 for additional information on file formatting for empirical weight-at-age.

10.3.15 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).

Control file continued:

Value	Label	Description
3	Spawner-	The options are:

Value	Label	Description
	Recruitment	1: null
	Relationship	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 $\ln(R_0)$, 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 \leq pop production
		8: Sheperd (3 parameters)

Read the required number of short parameter set-up lines (ex. LO HI INIT PRIOR PR_Type SD PHASE). These parameters are:

8.5	log(R0)	Log of virgin recruitment level
0.60	Steepness	Steepness of S-R, bound by 0.2 and 1.0 for the Beverton-Holt
COND:		If SRR = 5, 7, or 8
	3rd Parameter	Optional depending on which SRR function is used
0.60	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.
0	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.
8.5	log(R1)	Offset for initial equilibrium recruitment to virgin recruitment.
	AutoCorrelation	Autocorrelation in recruitment

Value	Label	Description
Then read additional spawner-recruitment conditions:		
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	<p>This factor determines what aspect of spawner-recruitment is affected by the environmental variable. The options are:</p> <ul style="list-style-type: none"> 1: annual deviations 2: R0 3: steepness <p>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 zeros 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.</p>
1	Do_Recr_Dev	<p>This selects the way in which recruitment deviations are coded:</p> <ul style="list-style-type: none"> 0: none (so all recruitments come from S-R curve) 1: dev vector (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.

Value	Label	Description
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 Recruitment Deviation Start Year: 0: skip (default) +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.
	6	Early Recruitment Deviation 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 Recruitment Deviations 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
	1970	First Year With Full Bias Adjustment
	2001	Last Year With Full Bias Adjustment
	2002	First Recent Year With No Bias Adjustment

Value	Label	Description
		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.
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		Minimum Recruitment Deviation: Min value for recruitment deviation. Default: -5
5		Maximum Recruitment Deviation: Max value for recruitment deviation. Default: 5
2		Number of Explicit Recruitment Deviations 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		
	<parameter line>	Full parameter line for each of the N periods of recruitment cycle
COND = If N explicit recruitment deviations is > 0, then enter N lines below		
	1977 3.0	Enter Year and Deviation

Value	Label	Description
1984	3.0	Two example 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

10.3.16 Spawner-Recruitment Function

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:

$$R_y = \frac{4hR_0SB_y}{SB_0(1-h) + SB_y(5h-1)} e^{-0.5b_y\sigma_R^2 + \tilde{R}_y} \quad \tilde{R}_y \sim N(0; \sigma_R^2) \quad (6)$$

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 is near 0.0.

Ricker

The Ricker Stock Recruitment curve is calculated as:

$$R_y = \frac{R_0SB_y}{SB_0} e^{h(1-SB_y/SB_0)} e^{-0.5b_y\sigma_R^2 + \tilde{R}_y} \quad \tilde{R}_y \sim N(0; \sigma_R^2) \quad (7)$$

Hockey-Stick

The hockey-stick recruitment curve is calculated as:

$$R_y = R_{min}R_0 + \frac{SB_y}{hSB_0}(R_0 - R_{min})(join) + R_0(1 - join) \quad (8)$$

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

$$join = \left[1 + e^{1000 * \frac{(SB_0 - hSB_y)}{SB_0}} \right]^{-1} \quad (9)$$

Survivorship

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

$$R_y = e^{\left(-z_0 + (z_0 - z_{min}) \left(1 - (SB_y/SB_0)^\rho\right)\right)} \quad \tilde{R}_y \sim N(0; \sigma_R^2) \quad (10)$$

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:

$$h = 0.2e^{z_0 z_{frac}(1-0.2^\beta)} \quad (11)$$

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 Shepherd 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 at recruitment) may be implemented in the future. For now, this means that the Z is really a $Z \cdot \Delta t$.
- 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. $Z_max = Z_0 + S_frac \cdot (0.0 - Z_0)$
- Then for the current level of pup production (e.g. total population fecundity, aka “spawning biomass”):
 - $Z_y = (1 - (Pup_y/Pups_0)Beta) \cdot (Z_max - Z_0) + Z_0$

- So $R_y = Pupy * exp(-Z_y)$
- 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.
 - $Pups_equil = Pups_0 * (1 - (LN(1/PPR) - Z_0) / (Z_max - Z_0))^{(1/Beta)}$
 - Then, $R_equil = Pups_equil * exp(-(1 - (Pups_equil/Pups_0)^{Beta}) * (Z_max - Z_0) + Z_0)$
- Code for the survival based recruitment is shown below:

```

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;
}

```

Shepherd

The Shepherd stock recruit curve is calculated as:

$$R_y = \left(\frac{SB_y}{SB_0} \right) \frac{5hR_0SB_0^c(1-0.2^c)}{(1-h_{adj}0.2^c) + (5h_{adj}-1)SB_y^c} e^{-0.5b_y\sigma_R^2 + \tilde{R}_y} \quad \tilde{R}_y \sim N(0; \sigma_R^2) \quad (12)$$

where c is the shape parameter for the stock recruitment curve, and h_{adj} is the

transformed steepness parameter defined as:

$$h_{adj} = \frac{(0.2 + (h - 0.2))(1 - 0.2(5 - 0.2^c))}{4 * 0.2^c} \quad (13)$$

10.3.17 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 deviation 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 deviations in early years against a preponderance of positive deviations 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.

10.3.18 Recruitment Likelihood

In SS2, recruitment $\log(L)$ contained a term, $+ N_forecast_rec_devs * \log(\sigma_R)$. This meant that the total $\log(L)$ changed according to how many forecast years were included in the model scenario. Worse, if σ_R was allowed to be estimated by SS2, then this term would cause all the zero deviations during the forecast period to drag the overall estimated value of σ_R 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 $\log L$ is equal to: $dev^2 / (2.0 * \sigma_R * \sigma_R) + offset * \log(\sigma_R)$; 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, σ_R may be estimable when there is good data to establish the time series of recruitment deviations. In the likefish example, turning on estimate of σ_R results in an estimated value that is very close to the root mean squared error (rmse) of the estimated recruitment deviations.

10.3.19 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. σ_R represents the true variability of recruitment in the population. It provides the constraining penalty for the estimates of recruitment

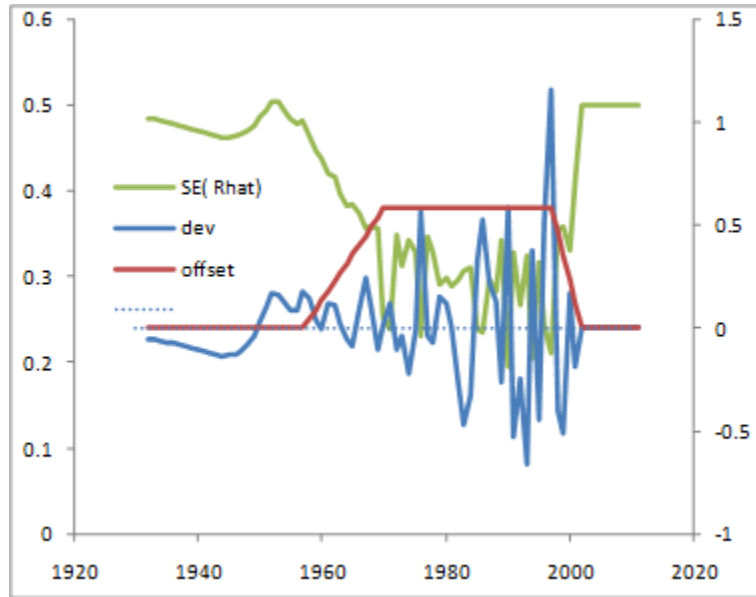
deviations and it is not affected by data. Where data that are informative about recruitment deviations are available, the total variability in recruitment, σ_R , is partitioned into a signal (the variability among the recruitment estimates) and the residual, the variance of each recruitment estimate (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 σ_R . Conversely, where there highly informative data about the recruitment deviations, then the variability among the estimated recruitment deviations will approach σ_R 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.

$$SE(\hat{r}_y)^2 + SD(\hat{r})^2 = \left(\left(\frac{1}{\sigma_d^2} + \frac{1}{\sigma_R^2} \right)^{-1/2} \right)^2 + \left(\frac{\sigma_R^2}{(\sigma_R^2 + \sigma_d^2)^{1/2}} \right)^2 = \sigma_R^2 \quad (14)$$

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

$$b_y = \frac{E\left(SD(\hat{r}_y) \right)^2}{\sigma_R^2} = 1 - \frac{SE(\hat{r}_y)^2}{\sigma_R^2} \quad (15)$$

of the offset then depends upon the partitioning of σ_R into between and within recruitment variability. The most appropriate degree of bias adjustment can be approximated from the relationship among σ_R , 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 σR^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.

10.3.20 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.

10.3.21 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 R_0 , 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 $N_{cycles} * R_0$. 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:

$$\begin{aligned} X &= \sum(e^p) \\ Y &= N_{cycle} \\ Penalty &= 100000 * (X - Y)^2 \end{aligned} \tag{16}$$

10.3.22 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.

10.3.23 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.

Control file continued:

Value	Description
0.2	F ballpark

Value	Description
	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 disable F ballpark
-3	F Method 1 = Pope's 2 = Continuous F as a parameter 3 = Hybrid F
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 the F method	
COND = 1: No additional input for Pope's approximation	
COND = 2: Continuous F	
0.10	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: Hybrid F	
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.
If N for F detail is > 0	
1 1980 1 0.20	fleet, year, season, 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.
0.05 4	

10.3.24 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.

10.3.25 Catchability

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

1. Do_Power

- (a) 0 = skip, so the survey is directly proportional to abundance (typical)
- (b) 1 = establish a parameter for non-linearity in survey-abundance linkage

2. Do_Env_Link

- (a) 0 = skip, no environmental on Q (typical)
- (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: $\ln(q_y) = \ln(q_{base}) + Q_{env_link_para} * Env_Value_y$.

3. Do_extra SD

- (a) 0 = skip (typical)
- (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 deviation 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 deviations $((1 + (\sum(devs))^2)^2 - 1)$ 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:

#Do_power (Den_depend)	Env-Var	Extra SD	Q type	Q offset	Fleet
1	0	1	2	0	#Fishery 1
0	0	1	2	0	#Fishery 2
0	0	0	4	0	#Survey 1
0	0	1	2	0	#Survey 2

Q types

- <0 = mirror another fleet
- 0 = float no bias adjustment
- 1 = float bias adjustment
- 2 = parameter no bias adjustment
- 3 = parameter with random deviations

- 4 = parameter with random walk
- 5 = mean unbiased float assignment to parameter

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 deviation, or to just read a template.

#Value	Label	Label, Description, and Options
1	Random effects	<p>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;</p> <p>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.</p>

For each positive element in columns for the catchability (Q) setup 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 standard deviation, 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:

#Lo	Hi	Init	Prior	PR_type	SD	Phase	Label
0	3	1	1	0	0.10	3	#Fishery1 power
0	0.5	0.10	0.10	0	0	4	#Fishery1 extra SD
0	0.5	0.10	0.10	0	0	4	#Fishery2 extra SD
0	0.5	0.10	0.10	0	0	4	#Survey2 extra SD
-7	5	0.50	0.50	-1	1	4	#Fishery1 base Q
-7	5	0.50	0.50	-1	1	4	#Fishery2 base Q

#Lo	Hi	Init	Prior	PR_type	SD	Phase	Label
-7	5	0.50	0.50	-1	1	4	#Survey1 base Q
0	1	0	0	1	0.10	3	#Survey1 Qrandwalk obs2
0	1	0	0	1	0.10	3	#Survey1 Qrandwalk obs3
0	1	0	0	1	0.10	3	#Survey1 Qrandwalk obs4
...
-7	5	0.50	0.50	-1	1	4	#Survey3 base Q

10.3.26 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.

#Example Setup for Size Selectivity Types				
#Pattern	Discard	Male	Special	Label
27	0	0	3	#Fishery1
1	0	0	0	#Survey1
0	0	0	0	#Survey2
#Age Selectivity Types				
11	0	0	3	#Fishery1
11	0	0	0	#Survey1
11	0	0	0	#Survey2

10.3.27 Selectivity Patterns

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

SIZE SELECTIVITY		
Pattern	N Parameters	Description
0	0	Selectivity equals 1.0 for all sizes
1	2	Logistic
2	8	Discontinued: Double logistic with defined peak (uses IF joiners). Use pattern #8 instead.
3	6	Discontinued
4	0	Discontinued: Set size selectivity equal to female fecundity. Use pattern #30 instead.
5	2	Mirror another selectivity. The two parameters select bin range.
6	2 + special value	Non-parametric
7	8	Discontinued: Double logistic with defined peak, uses smooth joiners; special = 1 causes constant selectivity above Linf for morph 1. Use pattern #8.

Pattern	N Parameters	Description
8	8	Double logistic, with defined peak, uses smooth joiners; special=1 causes constant selectivity above Linf for morph 1.
9	6	Simple double logistic with no defined peak.
15	0	Mirror another selectivity (same as for age selectivity).
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.
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/B0).

Notes on Special Selectivity Options:

- Do not input any size/age composition data 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 selectivity = 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 minimum age to 1.0.
11	2	Pick min-max age
12	2	Logistic
13	8	Double logistic, IF joiners. Use discouraged. Use pattern #18 instead.
14	nages+1	Each age, value at age is $\frac{1}{1+\exp(-x)}$
15	0	Mirror another selectivity
16	2	Coleraine single Gaussian
17	nages + 1	Each age as random walk from previous age. For all ages in the population beginning with Amin = 1 for the fishery and 2 for the survey, there is a corresponding set of selectivity parameters for each fleet, p_a . Click here for more information.
18	8	Double logistic, with defined peak, uses smooth joiners.
19	6	Simple double logistic with no defined peak.
20	6	Double normal with defined initial and final level. Recommended option. Test using SELEX-24.xls.
26	3	Exponential logistic
27	3+2*N_nodes	Cubic Spline

10.3.28 Selectivity Pattern Details

Pattern #1 (size) and #12 (age) - Simple Logistic

- p1 - size/age at inflection
- p2 - width for 95% selection; a negative width causes a descending curve.

Within SS logistic selectivity for the primary gender (if selectivity varies by gender) is formulated as:

$$S_l = \frac{1.0}{1 + \exp(-\ln(19)(L_l - p1)/p2)} \quad (17)$$

where L_l is the length bin. If age based selectivity is selected then the length bin is replaced by the age vector. If gender specific selectivity is specified the non-primary gender the p1 and p2 parameters are estimated as offsets. Note that with a large p2 parameter, selectivity

may not reach 1.0 at the largest size bin.

Pattern #5 (size) - Mirror Selectivity

Two 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 a value of 1 (e.g. first bin). If second parameter has value ≤ 0 , then interpreted as nlength (e.g. last bin). The source pattern must have a lower type number.

Pattern #6 (size) - Non-parametric selectivity

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.

Pattern #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 – INIT: selectivity at lengthbin=1 (minL) or age=0. 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) selectivity
- p8 – PEAKWIDTH: in width of flattop.

Pattern #14 (age) - Revise Age

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. SS calculates the age-based selectivity as:

$$temp = 9 - \max(p(1 : (nages + 1)))$$

$$S_a = \frac{1}{1 + \exp(-(p(a + 1) + temp))} \quad (18)$$

Pattern #17 (age) - Random Walk

This selectivity pattern provides for a random walk in $\ln(\text{selectivity})$. 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 $\ln(\text{selectivity})$ 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)=mexp(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;
}

```

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 (size) - 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: $\ln(\text{variance})$ on ascending side
- p4 – downslope: $\ln(\text{variance})$ on descending side

Pattern#23 (size) and #24 (size) - Double Normal Selectivity

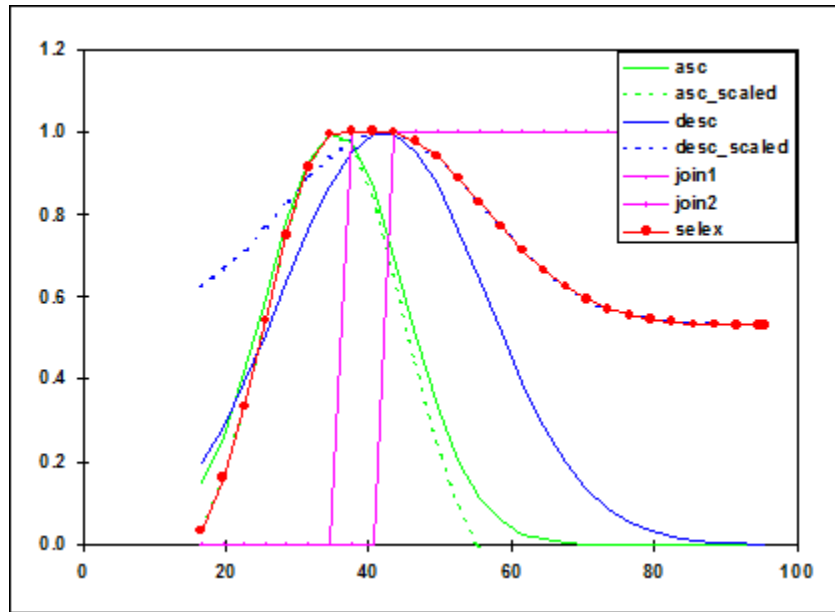
- p1 – PEAK: beginning size for the plateau (in cm)
- p2 – TOP: width of plateau, as logistic between PEAK and MAXLEN
- p3 – ASC-WIDTH: parameter value is $\ln(\text{width})$
- p4 – DESC-WIDTH: parameter value is $\ln(\text{width})$
- p5 – INIT: selectivity at first bin, as logistic between 0 and 1.
- 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.

Notes for Double Normal Selectivity:

- See spreadsheet SELEX-24.xls for parameterization example.
- 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.

Selectivity pattern #24, double normal, showing sub-functions and steep logistic joiners:



Pattern #15 (age) - Mirror

No parameters. Whole age range is mirrored from a user-specified fleet.

Pattern #16 - Gaussian (similar to Coleraine)

- p1 – age below which selectivity declines
- p2 – scaling factor for decline

Pattern #9 (size) and #19 (age) - 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.

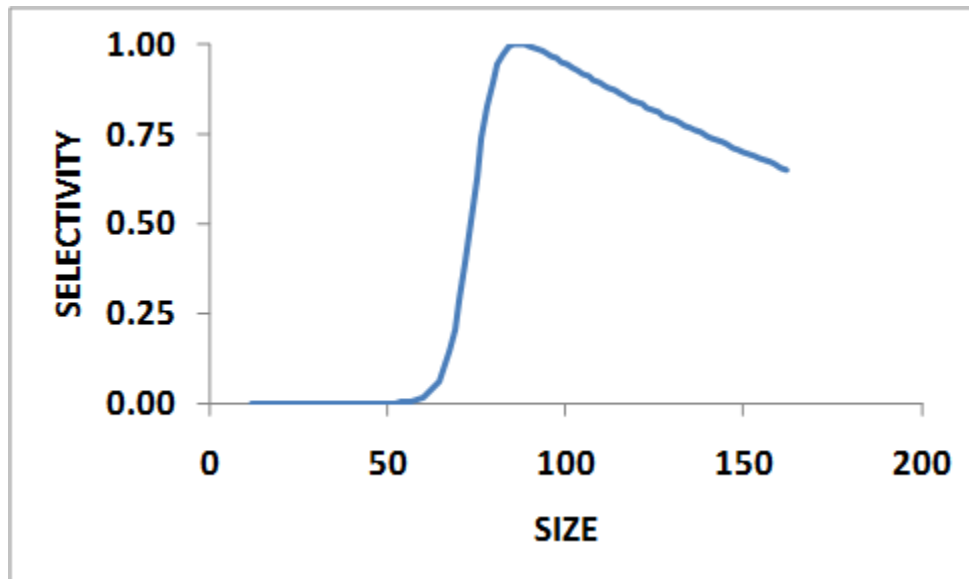
A value of 1.0e-6 is added to the selectivity for all ages, even those below the minage.

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.

$$\frac{e^{p3*p1(p2'-size)}}{1 - p3(1 - e^{p1(p2'-size)})} \quad (19)$$

Example: Exponential logistic selectivity with p1 = 0.30, p2 = 0.50, and p3 = 0.02:



Pattern #27 (size and age)- Cubic Spline

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 $\ln(\text{selectivity})$.

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.

Cubic spline selectivity is implemented within SS using the following code:

ADMB Code for cubic spline in SS

```

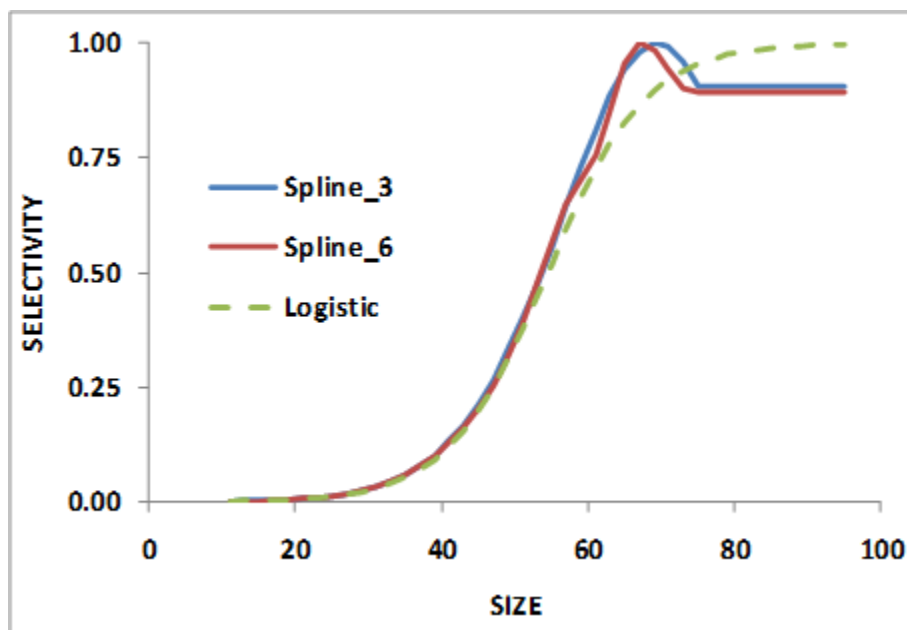
k=seltype(f,4); // N nodes

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_1 = splinefn(len_bins_m); // interpolate selectivity at the mid-
point of each population size bin
temp=max(tempvec_1(1,j2));
tempvec_1-=temp; // rescale to get max of 0.0
tempvec_1(j2+1,nlength) = tempvec_1(j2); //set constant above last node
sel = mfexp(tempvec_1);

```

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	PR	PR_TYPE	SD	PHASE	...	#Label
0	2	2.0	0	-1	0	-99	0	#SizeSpline Code
-0.001	1	0.13	0	1	0.001	3	0	#SizeSpline GradLo
-1	0.001	-0.03	0	0	0.001	3	0	#SizeSpline GradHi
11	95	38	0	-1	0	-99	0	#SizeSpline Knot1
11	95	59	0	-1	0	-99	0	#SizeSpline Knot2
11	95	74	0	-1	0	-99	0	#SizeSpline Knot3
-9	7	-3	0	1	0.001	2	0	#SizeSpline Value1
-9	7	-1	0	-1	0.001	99	0	#SizeSpline Value2
-9	7	-0.78	0	1	0.001	2	0	#SizeSpline Value3

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 necessary or 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:

- all parameter phases are adjusted by +1 so that only R0 is active in phase 1
- 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.

10.3.29 Retention

Retention is defined as a logistic function of size and can be dome-shaped. It does not apply to surveys. Seven parameters are used:

- p1 – ascending inflection
- p2 – ascending slope
- p3 – asymptotic retention (often a time-varying quantity to match the observed amount of discard)
- p4 – male offset to ascending inflection (arithmetic, not multiplicative)
- p5 – descending inflection
- p6 – descending slope
- p7 – male offset to descending inflection (arithmetic, not multiplicative)

$$Retention = \left(\frac{P3}{1 + e^{\frac{-(L - (P1 + P4 * male))}{P2}}} \right) * \left(1 - \frac{1}{1 + e^{\frac{-(L - (P5 + P7 * male))}{P6}}} \right) \quad (20)$$

10.3.30 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:

$$\text{deadfish} = \text{selex} * (\text{retain} + (1.0 - \text{retain}) * \text{discmort})$$

If discmort is 1.0, all selected fish are dead; if discmort is 0.0, only the retained fish are dead. Seven parameters are used:

- p1 – descending inflection
- p2 – descending slope
- p3 – asymptotic mortality
- p4 – male offset to descending inflection (arithmetic, not multiplicative)
- p5 – ascending inflection
- p6 – ascending slope
- p7 – male offset to ascending inflection (arithmetic, not multiplicative)

Discard mortality is calculated as:

$$\text{Mortality} = \left(1 - \frac{1 - P3}{1 + e^{\frac{-(L - (P1 + P4 * \text{male}))}{P2}}} \right) * \left(\frac{1}{1 + e^{\frac{-(L - (P5 + P7 * \text{male}))}{P6}}} \right) \quad (21)$$

10.3.31 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 $\exp(\text{this sum})$ per previous transform
- male parm 3 is added to the fourth selectivity parm (width of descending side); then $\exp(\text{sum})$ per previous transform
- male parm 4 is added to the sixth selectivity parm (selectivity at final size bin); then $1/(1+\exp(-\text{sum}))$ 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.

10.3.32 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:

1. Size selectivity for fishery 1
2. Retention for fishery 1
3. Male offsets for size selectivity for fishery 1
4. <repeat for additional fleets and surveys>
5. Age selectivity for fishery 1
6. Male offsets for age selectivity for fishery 1
7. <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 Environmental Linkage	
COND		If custom=0, then read one parameter line below and apply to all env functions; 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.		

Value	Label		Description
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 Method	Adjust	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.

10.3.33 Tag Recapture Parameters

Specify if tagging data are being used:

Value	Label		Description					
1	Tagging Data Present		0 = no read 1 = read following lines					
COND = 1								
#LO	HI	INIT	PRIOR	PR_TYPE	SD	PHASE	...	LABEL
-10	10	9	9	1	0.001	-4	0	#TG loss init 1
-10	10	9	9	1	0.001	-4	0	#TG loss init 2
-10	10	9	9	1	0.001	-4	0	#TG loss init 3
-10	10	9	9	1	0.001	-4	0	#TG loss chronic1

-10	10	9	9	1	0.001	-4	0	#TG loss chronic2
-10	10	9	9	1	0.001	-4	0	#TG loss chronic3
1	10	2	2	1	0.001	-4	0	#TG loss overdispersion1
1	10	2	2	1	0.001	-4	0	#TG loss overdispersion2
1	10	2	2	1	0.001	-4	0	#TG loss overdispersion3
-10	10	9	9	1	0.001	-4	0	#TG report fleet1
-10	10	9	9	1	0.001	-4	0	#TG report fleet2
-4	0	0	0	0	2	-4	0	#TG report decay1
-4	0	0	0	0	2	-4	0	#TG report decay2

The tagging reporting rate parameter is transformed within SS during estimation to maintain a positive value and is reported according to the transformation:

$$Tagging_Reporting_Rate = \frac{e^{input_parameter}}{1 + e^{input_parameter}} \quad (22)$$

10.3.34 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.

Variance Adjustment Factors		
-9999 1 0 No variance adjustment factors applied		
If variance adjustment factors are to be applied:		
Factor	Fleet	Value
1	2	0.5
4	1	0.25
4	2	0.75

Survey CV - Factor 1

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 - Factor 2

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 Weight - Factor 3

The 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 multiplied by the observation to get the adjusted standard deviation of the observation.

Length Composition - Factor 4

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

Age Composition - Factor 5

Age composition is treated the same way as length composition.

Size-at-Age - Factor 6

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.

Generalized Size Comp - Factor 7

Not yet implemented

Usage Notes

- 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.
- 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.

10.3.35 Lambdas (Emphasis Factors)

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

Value	Description
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 $+\log(\text{stddev})$ 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.

#Then read that number of lines containing the change information:				
#Component	Fleet/Survey	Phase	Lambda	SizeFreq Method
1	2	2	1.5	1
4	2	2	10	1
4	2	3	0.2	1
-9999	1	1	1	1

The codes for component are:

- 1 = survey;
- 2 = discard;
- 3 = mean weight;
- 4 = length;
- 5 = age;
- 6 = size frequency;
- 7 = size-at-age;
- 8 = catch;
- 9 = initial equilibrium catch;
- 10 = recruitment deviations;
- 11 = parameter priors;
- 12 = parameter deviations;
- 13 = crash penalty;
- 14 = morph composition;v 15 = tag composition;
- 16 = tag negbin.

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

#Lambdas (for information 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-at-at-age: 1
# 1	1	1	1	#size-at-at-age: 2
# 0	0	0	0	#size-at-at-age: 3
# 1	1	1	1	#init_equil_catch
# 1	1	1	1	#recruitments
# 1	1	1	1	#parameter priors
# 1	1	1	1	#parameter dev vectors
# 1	1	1	1	#crash penalty lambda

10.3.36 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 = no additional std dev reporting, 1 = read values							
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	Nselex	Growth	Ngrowth	Area for	Year for	N ages to
			Bins	Pattern	Areas	Natage	Natage	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 birth season 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.

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 n-at-age std bin picks (-1 in first bin to self generate)
--------------	------------------------------------------------------------------------

So a complete input looks like:

1	# 0 = no additional input, 1 = read stdev reporting lines								
COND > 0									
1	1	-1	5	1	5	1	-1	5	
5	16	27	38	46	#Vector with selex std bin picks				
1	2	14	26	40	#Vector with growth std bin picks				
1	2	14	26	40	#Vector with n-at-age std bin picks				

End Control File Input

999 #End of file

10.3.37 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 deviations.

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 deviations.

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.

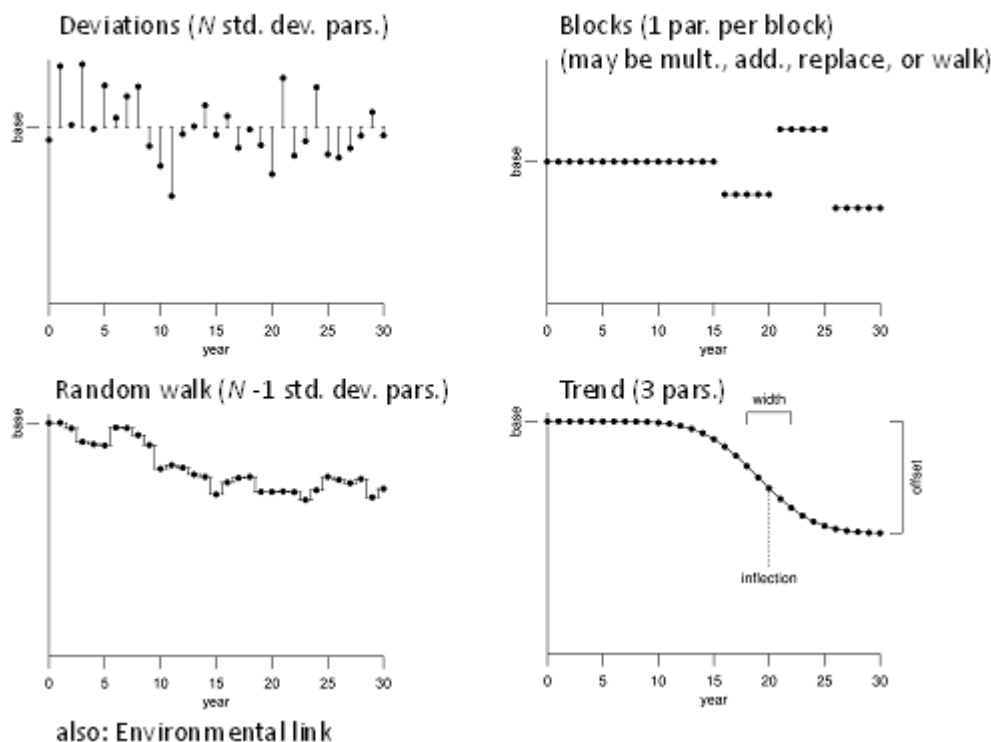
The available options for time-varying parameters are described below:

- Env Var - Element 8 in parameter setup
 - >0: multiplicative
 - <0: additive
 - abs(value): environmental index
- Use Dev - Element 9 in parameter setup
 - 1: multiplicative
 - 2: additive
 - 3: additive random walk
- Dev minyr - Element 10 in parameter setup
 - Year for deviations to start for parameter
- Dev maxyr - Element 11 in parameter setup
 - Year for deviations to end for parameter
- Dev StdDev - Element 12 in parameter setup
 - Standard deviation for deviations
- Blocks - Element 13 in parameter setup
 - >0: block index for parameter
 - <0: trend
- Block Functional Form: Element 14 in parameter setup
 - 0: multiplicative
 - 1: additive
 - 2: replace
 - 3: random walk

Example of location for specifying time-varying parameters

Env Var	Use Dev	Dev Minyr	Dev Maxyr	Dev StDev	Blocks	Block Fxn
>0: mult	1:mult	1973	1989	0.40	block index	0:mult

Time-varying parameter options



ENV	<p>A positive value, g, causes SS to set the annual working value of this parameter equal to a multiplicative function of Environmental Variable</p> $g:\text{parm}'(y) = \text{parm} * \exp(\text{link} * \text{env}(y,g))$ <p>A negative value, g, causes SS to set the annual working value of this parameter equal to a additive function of Environmental Variable g:</p> $\text{parm}'(y) = \text{parm} + \text{link} * \text{env}(y,-g)$ <p>Where, link is the environmental link parameter, parm is the base parameter being adjusted, parm' is the value after adjustment, and $\text{env}(y,g)$ 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 $\text{custom}=0$, then one short parameter line is used to define the min, max, init, etc, for each of the link parameters. If $\text{custom}=1$, then a separate line is read for each.</p>
Use_Dev	<p>A value of 1 invokes multiplicative: $\text{parm}'(y) = \text{parm} * \exp(\text{dev}(y))$</p> <p>A value of 2 invokes additive: $\text{parm}'(y) = \text{parm} + \text{dev}(y)$</p>

A value of 3 invokes additive random walk: $\text{parm}'(y) = \text{parm}'(y-1) + \text{dev}(y)$

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	<p>Block: 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.</p> <p>Trend: 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:</p> <p>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)</p> <p>P2: inflection year; if HI value for the base parameter is >1.1, then use as year, else use as fraction of range $\text{styr} - \text{endyr}$</p> <p>P3: width of change (units of std.dev. of years)</p>
Block Type	<p>This selects the way in which the block parameter creates an offset from the base parameter.</p> <p>0 means that $\text{parm}' = \text{baseparm} * \exp(\text{blockparm})$</p> <p>1 means that $\text{parm}' = \text{baseparm} + \text{blockparm}$</p> <p>2 means that $\text{parm}' = \text{blockparm}$</p> <p>3 means that $\text{parm}' =$ is additive offset from previous block. Note that blocks must be contiguous to use this option properly.</p>

10.3.38 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:

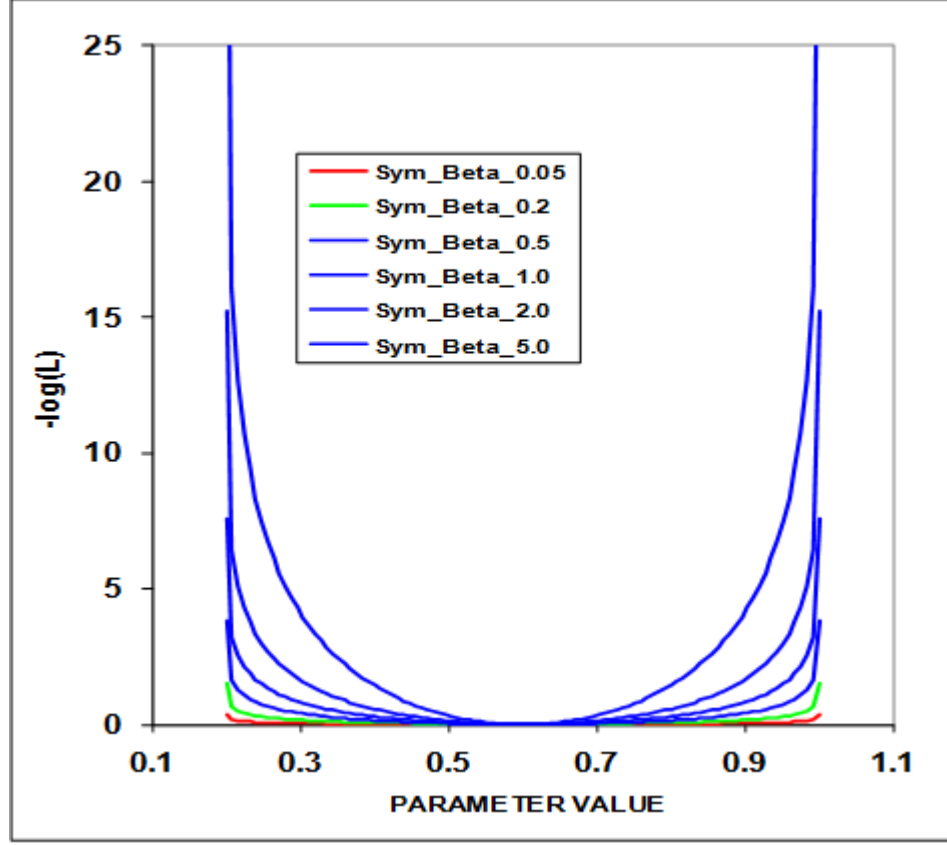
- -1 = No prior applied.
- 0 = Normal prior. Note that this function is independent of the parameter.

$$Pr_Like = 0.50 * (\frac{Pval - Prior}{Prior_SD})^2 \quad (23)$$

- 1 = Symmetric bet prior is scaled between parameter bounds, imposing a larger penalty near the bounds. Prior standard deviation of 0.05 is very diffuse and a value of 5.0 provides a smooth U-shaped prior.

$$\mu = -P_SD * (\log((Pmax + Pmin) * 0.5 - Pmin) - P_SD * (\log(0.5))) \quad (24)$$

$$Prior_Like = -(\mu + (P_SD * (\log(Pval - Pmin + Pconst)))) + (P_SD * (\log(1 - (\frac{(Pval - Pmin - Pconst)}{(Pmax - Pmin)})))) \quad (25)$$



Prior distributions for the symmetric beta distribution.

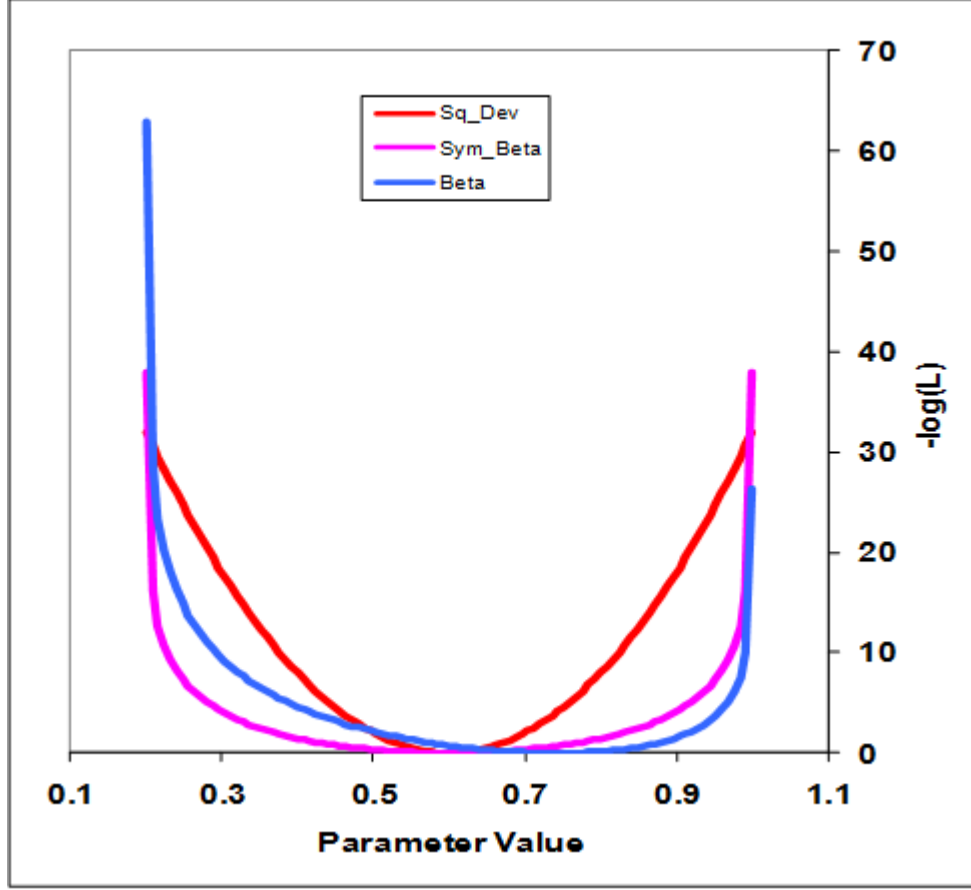
- 2 = Beta prior. The definition of μ is consistent with CASAL's formulation with the Bprior and Aprior corresponding to the m and n parameter.

$$\mu = \frac{Prior - Pmin}{Pmax - Pmin}$$

$$\tau = \frac{(Prior - Pmin)(Pmax - Prior)}{P_SD^2} - 1.0 \quad (26)$$

$$Bprior = \tau * \mu; Aprior = \tau(1.0 - \mu)$$

$$\begin{aligned} Prior_Like = & (1.0 - Bprior) * \log(Pconst + Pval - Pmin) + \\ & (1.0 - Aprior) * \log(Pconst + Pmax - Pval) - \\ & (1.0 - Bprior) * \log(Pconst + Prior - Pmin) - \\ & (1.0 - Aprior) * \log(Pconst + Pmax - Prior) \end{aligned} \quad (27)$$



Comparison of the symmetric beta and the beta prior functions

- 3 = Lognormal prior. Note that lower bound on the parameter must be >0.0 . The prior value is input into the parameter line in log space while the initial parameter value is defined in normal space (e.g. INIT = 0.20, PRIOR = -1.609438).

$$Prior_Like = 0.50 * \left(\frac{(\log(Pval) - Prior)}{Pr_SD} \right)^2 \quad (28)$$

- Pval is the value of the parameter for which a prior is being calculated, Pmin and Pmax are the bounds on the parameter, Prior is the value of the parameter prior, or the first of the 2 factors controlling the calculation of the prior, Pr_SD is the value of the prior's standard deviation, or the second of the 2 factors controlling the calculation of the prior, Pconst is a small constant (0.0001), and Prior_Like is the calculated value of the prior's contribution to the log likelihood.

11 Output Files

11.1 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 quantities 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.

11.2 Derived Quantities

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

11.2.1 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.

11.2.2 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 = $(1-\text{SPR})/(1-\text{SPR}_{\text{TGT}})$
- 2 = $(1-\text{SPR})/(1-\text{SPR}_{\text{MSY}})$
- 3 = $(1-\text{SPR})/(1-\text{SPR}_{\text{Btarget}})$
- 4 = raw SPR

11.2.3 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/F_{SPR}
- 2 = F/F_{MSY}
- 3 = F/F_{Btarget}

11.2.4 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.

11.2.5 MSY and other Benchmark 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

Benchmark Item	Description
TotYield_MSY	Total yield (biomass) at MSY
RetYield_MSY	Retained yield (biomass) at MSY

11.3 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.

11.4 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
3
# First year of projection (Yinit)
2002
# First Year of rebuilding period (Ydecl)
1999
# Number of simulations
1000
# Maximum number of years
500
# Conduct proejctions with multiple starting values (0 = No, 1 = Yes)
0
# Number of parameter vectors
1000
# 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)
1v # 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 0.000450117 0.00436298 0.0271371 <deleted values>
# Age specific selectivity and weight adjusted for discard and discard mortality
#wt and selex for gender, fleet: 1 1
0.146708 0.320119 0.555587 0.830467 <deleted values>
0.0122887 0.0351722 0.0838682 0.165479 <deleted values>
#wt and selex for gender ,fleet: 2 1
0.150944 0.33768 0.588317 0.874376 <deleted values>
0.0127241 0.0380999 0.0922667 <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>
1425.96 797.624 1234.77 428.207 <deleted values>
# gender = 2
0.1 0.1 0.1 0.1 0.1 <deleted values>
1425.96 797.531 1233.66 <deleted values>
# Age-structure at Ydeclare= 1999
598.671 652.739 2925.76 2227.69 <deleted values>
598.671 652.666 2923.27 2221.05 <deleted values>
# Year for Tmin Age-structure (set to Ydecl by SS) 1999
1999
# recruitment and biomass
# Number of historical assessment years
33
# Historical data
# year recruitment spawner in B0 in R project in R/S project
1970 1971 1972 1973 1974 1975 1976 <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 <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
3
# Process for overriding (-1 for average otherwise index in data list)
2000 1 2000
2001 1 2001
2002 1 2002
# Which probability to product detailed results for (1=0.5; 2=0.6; etc.)
3
# Steepness sigma-R Auto-correlation
0.610789 0.6 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
# Maximum possible F for projection (-1 to set to FMSY)
-1
# Defintion of recovery (1=now only; 2=now or before)
2
# Projection type
4
# 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
# File with multiple parameter vectors
rebuild.SS0
# User-specific projection (1=Yes); Output replaced (1->9)
0 5

```

```

# Catches and Fs (Year; 1/2/3 (F or C or SPR); value); Final row is -1v 2002 1 1
-1 -1 -1
# Split of Fs
2002 1
-1 1 1 1
# Yrs to define TTARGET for projection type 4 (aka 5 pre-specified inputs)
2011 2012 2013 2014 2015 2016 2017 2018
# 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
# CV of implementation error
0

```

11.5 Bootstrap Data Files

Data.ss_new: contains a user-specified number of data files, 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.

11.6 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 end year. 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:

- Cohorts continue growing according to their specific growth parameters in the forecast period rather than staying static at the endyr values.
- 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 F_{SPR} and the search for F_{MSY} 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 F_{MSY} 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_unfis hed(R0)	--	1.000	1.000	8853		
SPB_unfis hed(S0)	--	7.193	7.193	63680		
BIO_Smry_unfis hed	--	18.493	18.493	163727		
+	+	+	+	+		
SPR_target	0.500					
SPR_calc	0.500					
Fmult	0.261					
Exploit(Y/Bs mry)	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/Bs mry)	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/Bs mry)	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_fra																
tion_of_logbias_adj_before_endyr+1:																
and_this_value_after_endyr:																
Extra_emphasis_on_forecast_reorde																
s_before_endyr+1:																
N_forecast_yrs_and_with_stddev																
OY_Control:																
+																
+																
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																
pop																
+																
Annual_all_area_values_stored_in_sdreport_vector'depletion'_beginning_in_element_8																
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											

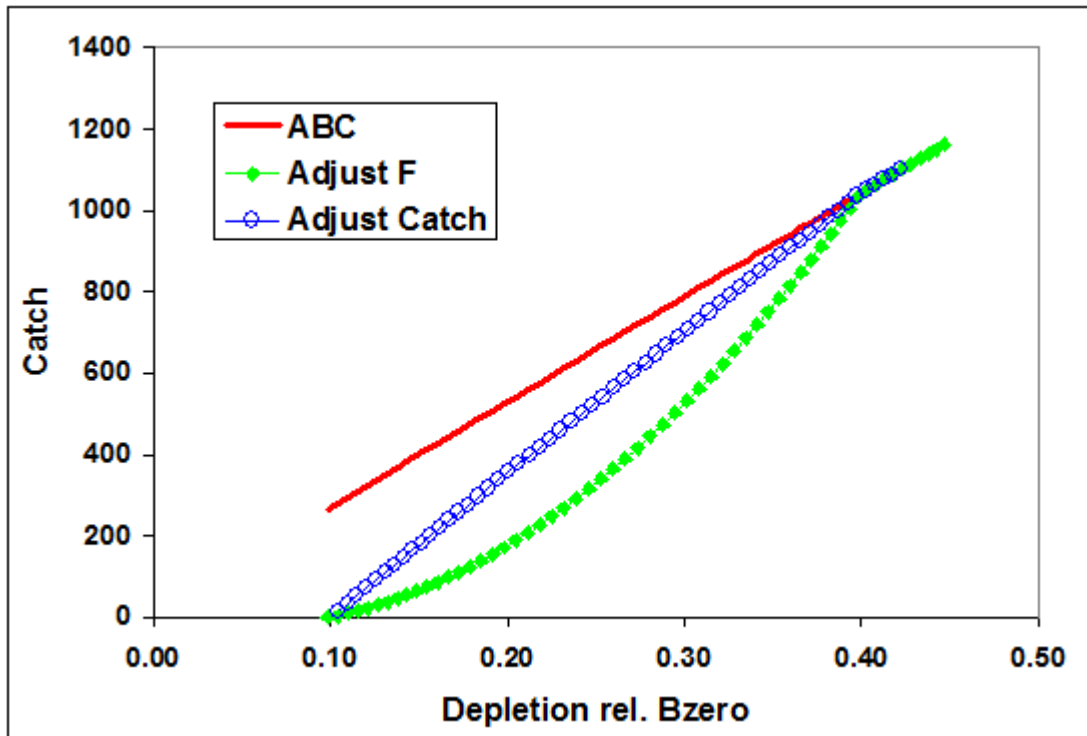
where:

- 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:

- depletion level in end year
- depletion level in end year+1
- MSY (if calculated, else spbio in endyr-1)
- BMSY (if calculated, else spbio in endyr)
- SPRMSY (if calculated, else spbio in endyr+1) then the time series of:

- Spawning biomass
- Recruitment
- Depletion level
- 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)
- Total exploitation rate (total dead catch divided by the summary biomass at the beginning of the year).



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

11.7 Main Output File, report.sso

This is the primary output file. Its major sections are listed below.

The sections of the output file are:

- SS version number with date compiled. Time and date of model run. This info appears at the top of all output files.
- Comments
 - Input file lines starting with #C are echoed here
- Keywords

- List of keywords used in searching for output sections.
- Fleet Names
 - List of fishing fleet and survey names assigned in the data file
- Likelihood
 - Final values of the negative log(likelihood) are presented.
- Input Variance Adjustments
 - The matrix of input variance adjustments is output here because these values affect the logL calculations
- Parameters
 - 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.
- Derived Quantities
 - This section starts by showing the options selected from the starter.ss and forecast.ss input files:
 - * SPR ratio basis
 - * F report basis
 - * B ratio denominator

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.

- MGparm by year after adjustments
 - This block shows the time series of Mgpars by year after adjustment by environmental links, blocks and deviations.
- SELparm (size) by year after adjustments
 - This block shows the size selectivity parameters, after adjustment, for each year in which a change occurs.
- SELparm (age) by year after adjustments
 - This block shows the age selectivity parameters, after adjustment, for each year in which a change occurs.
- Recruitment Distribution
 - This block shows the distribution of recruitment across growth patterns, genders, birthseasons, and areas in the endyr of the model.
- Morph Indexing
 - 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.
- Size Freq Translation
 - 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.
- Movement
 - This block shows movement rate between areas in a multi-area model.
- Exploitation

- 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 .
- 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 gaging 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 $\ln(N_{t+1,a+1}N_{t,a})$. 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.

12 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.

12.1 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%
```

12.1.1 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

```

12.1.2 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)

```

12.1.3 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
```

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.

12.1.4 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.

12.1.5 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 iteration. So if 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.

12.1.6 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.

12.1.7 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) To start the MCMC chain from specific values change the par file; run the model with estimation, adjust the par file to the values that the chain should start from, change within the starter file for the model to begin from the par file, and call the MCMC function using `ss3 -mcmc xxxx -mcsave yyyy -nohess -noest`.
- (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.

12.1.8 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.google.com/p/r4ss which facilitates 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).
```

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 by the `SS_output` function:

```

RGui - [R Console]
File Edit View Misc Packages Windows Help

[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.ct1"

$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          1306.519999999999818
Catch           0.0000000000431523
Equil_catch      0.00000000000000000 1
Survey          -3.9812099999999999
Length_comp     651.0109999999999673
Age_comp        586.6159999999999854
Size_at_age     79.2207999999999970
Recruitment     -7.39975000000000000 1
Forecast_Recruitment 0.00000000000000000 1
Farm_priors     1.05577000000000001 1
Farm_softbounds 0.00120521000000000 NA
Farm_devs       0.00000000000000000 1
Crash_Pen       0.00000000000000000 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

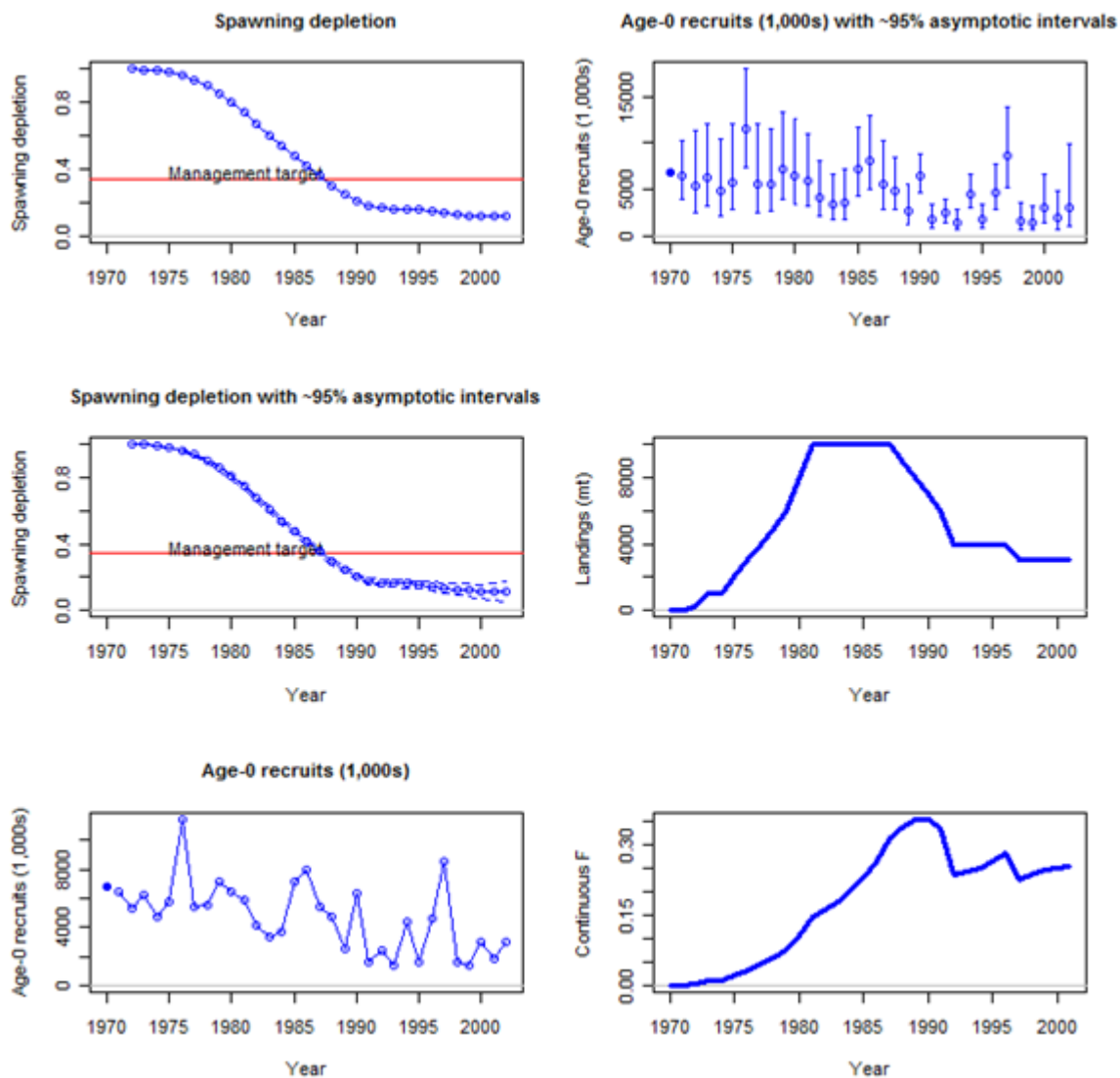
$stable_of_phases

-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 the plots created using the SS_plots function:



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

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
