Transitioning to SS 3.30

Contents

[Introduction 2](#_Toc433025053)

[Structural change in timing calculations 3](#_Toc433025054)

[Biology Parameter Order 4](#_Toc433025055)

[conditional read of 3.24 vs 3.30 input format 6](#_Toc433025056)

[Generic fleet order 7](#_Toc433025057)

[Catch Data format 8](#_Toc433025058)

[Catch multiplier 9](#_Toc433025059)

[Environmental variable linkages 10](#_Toc433025060)

[Survey of mgdev 11](#_Toc433025061)

[Re-Organize and add to the Survey and Q sections 12](#_Toc433025062)

[Dev vector variance as a model parameter 13](#_Toc433025063)

[Shepard SRR 14](#_Toc433025064)

[Restricted range for use of the ALK 15](#_Toc433025065)

[Composition likelihood option – Dirichlet multinomial 16](#_Toc433025066)

[Area-specific spawner-recruitment 17](#_Toc433025067)

[Bycatch fleets – more options for benchmark and forecast 20](#_Toc433025068)

# Introduction

This document provides an overview of the major structural change introduced in Stock Synthesis version 3.30, and various features made available in this release. The structural change introduces finer temporal resolution within each year or season and resulted in a major rebuild of the growth module. With this change, size-at-age can now be calculated at finer intervals within a season, not just the beginning and end. This means that a model with annual time steps can now look at spring and fall surveys more accurately. The time at which observations occur will be entered as a real valued month, rather than an integer season, and the season within which an observation occurs will be calculated at runtime from the season duration set-up. The new temporal sequencing will now allow for an explicit time lapse between spawning and recruitment such that age 0 growth and mortality begins at the time of recruitment. The input file reader section of SS\_3.30 is backwardly compatible with SS\_3.24 formatted input files, with few exceptions.

The new features introduced in SS\_3.30 cover a variety of topics to better meet the needs of the user community. This evolution will continue. In addition, we are working on a graphical interface and a web-based communication hub to serve the SS community.

The core code for the ADMB version of SS is now ten years old and some good new ideas cannot easily be fit into this code base. Hence, while continuing to support and augment the SS\_3.30 system, we will also be building a more modular model with comparable and expanded capabilities to be faster, easier and more powerful. But for the next couple of years, enjoy SS\_3.30.

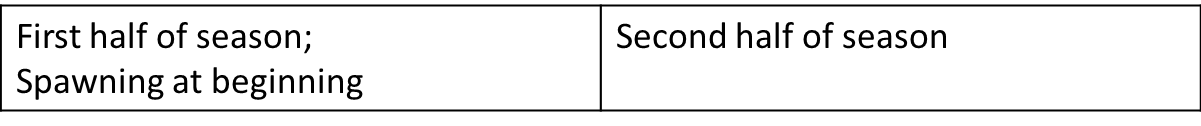
# Structural change in timing calculations

The timing of events in SS\_3.24 and earlier was structured as follows:

* Mortality was constant for entire season;
* Body weight-at-age for fisheries and surveys were calculated from growth to the midpoint of each season;
* Reproductive output was calculated using body size at the beginning of the specified spawning season.
* Survey timing is survey-specific and specified as a fraction of a season,
* Survey numbers-at-age calculated at survey timing interpolated using e-Z

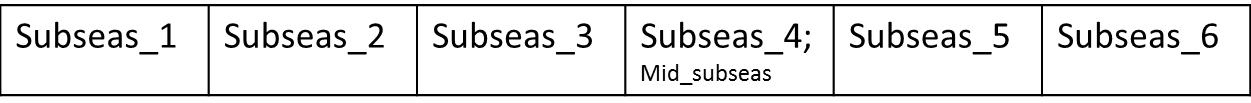
ALK

ALK



The timing of events in SS\_3.30 is as follows

* Continuous Z for entire season, so same as 3.24;
* even number (min = 2) of subseasons per season (regardless of season duration);
  + so 2 subseasons will mimic 3.24
  + a 6 subseason example is shown below
  + specifying more sub seasons will give finer temporal resolution, but will slow the model down, the effect of which is mitigated by only calculating growth as needed
* Survey timing is now cruise-specific and specified in units of months.fraction (e.g. Apr 15 = 4.5);
* Survey integer season and spawn integer season assigned at runtime based on real month and season duration(s);
* The closest subseason is calculated for each observation;
* Growth and ALK only calculated at beginning and mid\_subseason, or when there is an observation in that subseason (ALK\* in figure below);
* Fishery body weight and size composition uses mid\_subseas growth;
* Survey body weight and size composition calculated using the nearest subseason;
* Reproductive output now has specified spawn timing (in months.fraction) and interpolates growth to that timing;
* Survey numbers calculated at cruise survey timing using e-Z



ALK\*

ALK

ALK\*

ALK

## Biology Parameter Order

The weight-length and maturity-fecundity parameters are now specific to each growth pattern, so the order of parameters has changed. SS does this re-ordering when it reads a SS3.24 input file. It may not be able to read a 3.24 setup that already has multiple growth patterns.

Settlement Events

V3.24

* Recruitment happened at real age 0.0 at beginning of a season, including the spawning season
* Recruits distributed among areas, seasons, growth\_pattern

V3.30

* Recruitment happens in specified settlement events (Gpat, Month, Area);
* Unlike 3.24, there must be one settlement event specified (does not default to all born on Jan 1 in area 1 of growth pattern 1)
* Number of unique settlement timings calculated at runtime;
* Now there can be elapsed time between spawning and recruitment;
* Growth and natural mortality of the platoon begins at time of settlement, which is its real age 0.0 for growth;
* All fish become integer age 1 (for age determination) on their first Jan 1;
* Recruitment can occur >12 months after spawning

3 # number of recruitment settlement events

0 # year\_x\_area\_x\_settlement\_event interaction requested (only for recr\_dist\_method=1)

#GPat month area (for each settlement)

1 1 1

1 4 1

1 8 1

# conditional read of 3.24 vs 3.30 input format

Starter.ss currently ends with 999

SS 3.30 now interprets this 999 to mean that rest of input files are in 3.24 input format

Replace this 999 in starter.ss with 3.30, and SS will read forecast, data, and control files in 3.30 format

All ss\_new files are in 3.30 format, so starter.ss\_new has 3.30 on the last line

Some Mgparms are in new sequence, so SS 3.30 cannot read a ss3.par file produced by the 3.24 exe

# Generic fleet order

Previously, fishing fleets were listed first, followed by survey only fleets. Input of catch was only associated with fishing fleets and there was an option for a fishing fleet to be designated as "bycatch only" such that the input catch values were ignored. A problem with this approach is that addition or subtraction of a fishing fleet required renumbering the "fleet" ID on data for all higher numbered fleets and surveys.

Solution - All fleets now have the same status within the model structure and each has a specified fleet type.

Available fleet types are: catch fleet, bycatch only, survey. Future types can be: environment, predator, ignore.

Fleet input is rotated to row-orientation and now looks like:

3 #\_Nfleets (including surveys)

#\_fleet\_type: 1=catch fleet; 2=bycatch only fleet; 3=survey; 4=ignore

#\_survey\_timing: -1=for use of catch-at-age to override the month value associated with a datum

#\_fleet\_area: area the fleet/survey operates in

#\_units of catch: 1=bio; 2=num (ignored for surveys; their units read later)

#\_equ\_catch\_se: standard error of log(initial equilibrium catch)

#\_catch\_se: standard error of log(catch); can be overridden in control file with detailed F input

#\_need\_catch\_mult: new feature to create a fleet-specific, time-variable parameter to rescale catch

#\_fleetname

#\_rows are fleets; columns are: fleet\_type, timing, area, units, equ\_catch\_se, catch\_se, need\_catch\_mult, fleetname

1 0.5 1 1 0.01 0.01 0 FISHERY

3 0.5 1 2 0.1 0.1 0 Trawl\_Survey

3 0.5 1 2 0.1 0.1 0 Recruit\_Survey

# Catch Data format

The input format for catch data has changed. In the new format:

* Input is organized as a list, not as a table
* Each record in the list has the format: yr, seas, fleet, catch amount, catch se
* Only positive catches need to be entered, so there is no need for records for the survey fleets
* There is no longer a need to enter a value for the number of catch records, instead the list is terminated by entering a record with the value -9999 in the year field
* Initial equilibrium catch is now season specific
* There is no longer a separate input for the initial equilibrium catch. This is now entered along with the other catch records by specifying the year as -999
* As before:
  + If the season field is greater than nseasons, the catch record is added to the catch for season=nseasons;
  + If the season field has the value 0, then the catch for that record is added to all seasons of that year equally.

The new format 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| #Year | Season | Fleet | Landed Catch amount | Catch\_se |
| -999 | 1 | 1 | 56 | .05 |
| -999 | 2 | 1 | 62 | .05 |
| 1975 | 1 | 1 | 876 | .05 |
| 1975 | 2 | 1 | 343 | .05 |
| -999 | 1 | 2 | 55 | .05 |
| -999 | 2 | 2 | 22 | .05 |
| 1975 | 1 | 2 | 555 | .05 |
| 1975 | 2 | 2 | 873 | .05 |
| -9999 | 0 | 0 | 0 | .05 |

# Catch multiplier

Implement catch\_mult(y,f) as a fleet-specific multiplier of catch.

A flag in fleet\_setup indicates which fleets are requesting a catch\_multiplier.

It has year-specific, not season-specific time-varying capabilities.

Do this as a MGparm, so can inherit all time-varying characteristics of MGparms.

In the catch\_like calculation, expected catch is multiplied by catch\_mult(y,f) before being compared to the observed retained catch, so a value of 1.1 means that the observed catch has overestimated actual catch by 10%.

Also implement in Pope's and hybrid F calculations.

# Environmental variable linkages

Begin to expand the scope of environmental variable linkages.

// SS\_Label\_Info\_14.4.1.2 #Adjust for env linkage

// June 6 begin to add 2 parameter env linkages

// P1 will be the current "slope" and P2 will be a new offset

if(MGparm\_env(f)>0)

switch(MGparm\_envtype(f))

case 1: // exponential MGparm env link

mgp\_adj(f)\*=mfexp(MGparm(MGparm\_env(f))\*(env\_data(yz,MGparm\_envuse(f))-MGparm(MGparm\_env(f))));

case 2: // linear MGparm env link

mgp\_adj(f)+=MGparm(MGparm\_env(f))\*(env\_data(yz,MGparm\_envuse(f))-MGparm(MGparm\_env(f)));

case 3: // logistic MGparm env link

mgp\_adj(f)\*=2.00000/(1.00000 + mfexp(-MGparm(MGparm\_env(f))\*(env\_data(yz,MGparm\_envuse(f))-MGparm(MGparm\_env(f)))));

# Survey of mgdev

Special size selectivity type = 35 will invoke setting e(survey)=f(Mgparm\_dev(y))

But the link function, f, needs more options than simple unity!

# Re-Organize and add to the Survey and Q sections

# Dev vector variance as a model parameter

Currently the se of the dev is a constant in the invoking long parameter line, and the options are a dev vector or a random walk vector.

In 3.30, the se is now a parameter and there is a rho for the degree of mean reversion:

Mean-reverting random walk with parameters:

// =(1-rho)\*mean + rho\*prevval + dev // where mean = 0.0

MGparm\_dev\_rwalk(k,j)=MGparm\_dev\_rho(k)\*MGparm\_dev\_rwalk(k,j-1)+MGparm\_dev(k,j);

Where input will be something like:

# standard error parameters for first MG dev vector

0.1 0.1 0.1 0.1 0.1 0.1 0.1 # RecrDist\_Area\_1\_dev\_se

0.1 0.1 0 0.1 0.1 0.1 0.1 # RecrDist\_Area\_1\_dev\_rho #

# RecrDist\_Area\_1\_DEVadd\_1972 0 0 0 0 0 0 0 0

-5 #\_MGparm\_Dev\_Phase

# Shepard SRR

# Restricted range for use of the ALK

Create a tolerance value for restricting the range of calculations when addressing the length-at-age distributions. Search for the range of bins that meet this tolerance in the first call and between phases. Then subsequent use of the ALK will only occur over this age-specific range of length bins, e.g.

ALK\_idx=(s-1)\*N\_subseas+subseas; // timing within a year

ALK\_finder=(ALK\_idx-1)\*gmorph+g; // combined index for timing and platoon

ALK\_range\_use=calc\_ALK\_range(len\_bins,use\_Ave\_Size\_W,use\_SD\_Size,ALK\_tolerance);

ALK\_range\_g\_lo(ALK\_finder)=column(ALK\_range\_use,1);

ALK\_range\_g\_hi(ALK\_finder)=column(ALK\_range\_use,2);

ALK(ALK\_idx,g)=calc\_ALK(len\_bins,ALK\_range\_g\_lo(ALK\_finder),ALK\_range\_g\_hi(ALK\_finder),use\_Ave\_Size\_W,use\_SD\_Size);

optimal setting for the tail seems to be 0.0001

getting more iterations per second with more compression,

but more iterations are needed to converge

very big tails (0.05) are much faster, but don't get same biomass and logL

# Composition likelihood option – Dirichlet multinomial

Add option to allow for dirichlet multinomial likelihood

Option is invoked with a field in the composition data specification section

Creates a parameter

Allows mirroring

# Area-specific spawner-recruitment

5. Conceptual problem in equilibrium calcs if fish move between areas

Equil\_Calc takes some total recruitment (Equ\_Recr)

it distributes according to recr\_dist,then calculates equilibrium numbers at age taking into account F and movement between areas if outputs a value for total SPR (spawners per recruit) then SPR is input to Equ\_Spawn\_Recr\_Fxn to calculate the equilibrium total recruits and spawners from this SPR

but if recruitment to an area depends on that areas SPB, then SPR needs to be area-specific but with movement between areas, the SPB in an area depends on the movement so SPR cannot be calculated independently for each area if there is movement so equil\_calc will need to output area-specific SPR; this seems feasible but it is still somewhat circular because the distribution of recruits to areas in equil\_calc will depend on the SPB by area calculated by equil\_calc

6. Need to reconcile the area-aspect of recr\_dist with the R=F(SPB\_area) area-specific R0 global steepness movement mixes B between areas fishing reduces area-specific B reduced area-specific B reduces E(area-specific R)

does global R=E(sum B) equal R = sum (E(Ba)) ??

current approach:

1. Given some reference recruitment level - R

2. Call equil\_calc

a. distribute R to areas according to recr\_dist

b. loop areas within ages

c. for each age,/area calc SPB, Yield, Survivors

d. for each age mix survivors among areas according to movement specs

e. accumulate total SPB and Yield and return from equil\_calc

3. calc SPB/R and Yield/R

4. call Equ\_Spawn\_Recr\_Fxn to get B\_equil and R\_equil from SPB/R and SR parms

Proposed two-stage iterative area-specific approach for equilibrium calcs:

step A:

1. Given some reference recruitment level - R

2. Call equil\_calc

a. distribute R to areas according to unfished recr\_dist (which needs to be defined here)

b. loop areas within ages

c. for each age,/area calc SPB, Yield, Survivors

d. for each age mix survivors among areas according to movement specs

e. accumulate total and area-specific total SPB and Yield and return from equil\_calc

3. calc area-specific SPB/R and Yield/R, using area-specific R

4. call Equ\_Spawn\_Recr\_Fxn for each area to get B\_equil and R\_equil from SPB/R and SR parms

5. sum across areas to get adjusted total B\_equil and R\_equil

6. use ratio of B\_equils among areas to calculate adjustment to recr\_dist

step B:

7. call equil\_calc again, using the adjusted recr\_dist

General forecast changes

* Default forecast is now one year, not zero years

# Bycatch fleets – more options for benchmark and forecast

currently bycatch fleets must use F\_Method=2 and are excluded from the catch logL

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.

Add option so that bycatch only fleet:

* can have retained and discarded catch calculated normally, or
* all their catch will be assigned to discard

Because MSY and Yield per recruit are calculated in terms of dead catch, they currently include catch from bycatch fleets.  provide option such that:

* bycatch only fleets are treated normally in benchmark and forecast
* F for bycatch only fleets is kept constant in benchmark and forecast, so is not included in any forecast cap&allocation calculations.  It will not be part of ABC, but it will still be calculated and reported.  The level of F for bycatch only fleets will then need to be set as a constant, or calculated as a mean from a range of specified years.