Transitioning to SS 3.30

Contents

[Structural change in timing calculations 2](#_Toc424135163)

[Biology Parameter Order 3](#_Toc424135164)

[conditional read of 3.24 vs 3.30 input format 5](#_Toc424135165)

[seasons in initial year 6](#_Toc424135166)

[Generic fleet order 7](#_Toc424135167)

[Catch multiplier 8](#_Toc424135168)

[Survey of mgdev 9](#_Toc424135169)

[Q offset and other link fxns 10](#_Toc424135170)

[Dev vector variance as a model parameter 11](#_Toc424135171)

[Shepard SRR 12](#_Toc424135172)

[Restricted range for use of the ALK 13](#_Toc424135173)

[Area-specific spawner-recruitment 14](#_Toc424135174)

[Bycatch fleets – more options for benchmark and forecast 15](#_Toc424135175)

# 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.

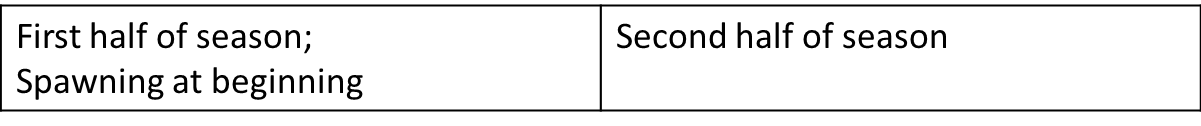
# Structural change in timing calculations

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

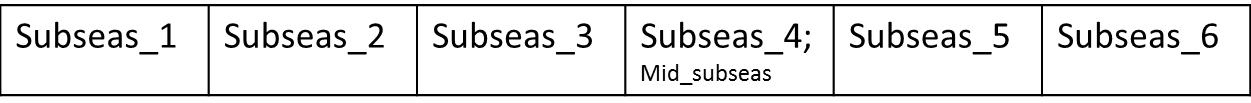
ALK

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

# seasons in initial year

Still can read 3.24 files in old format

New format is:

96 #\_N\_lines\_of\_catch\_to\_read

#\_catch\_biomass(mtons):\_columns\_are\_year,season,fleets(including surveys with no catch); -999 for initial equilibrium

-999 1 0 0 0 0 0

-999 2 0 0 0 0 0

-999 3 0 0 0 0 0

1971 1 0 0 0 0 0

1971 2 0 0 0 0 0

1971 3 0 0 0 0 0

1972 1 23.8468 21.8865 23.2409 0 0

1972 2 21.6049 21.3974 21.3956 0 0

1972 3 21.9649 24.2593 21.4618 0 0

# 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 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.

# 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!

# Q offset and other link fxns

# 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

# Area-specific spawner-recruitment

# 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.