

(IOTC-2008-WPM4-01 rev)

Please contact the editor if you want to cite this document.

Two software will be released after completed and approved by the developers.

NOTES ON TWO NEW ASPM SOFTWARE BASED ON “AD MODEL BUILDER” AND “SS3”

Edited by Tom Nishida

National Research Institute of Far Seas Fisheries (NRIFSF), Shimizu, Shizuoka, Japan

1. INTRODUCTION

ASPM software coded by FORTRAN was made by Victor Restrepo (ICCAT, 1997) which has been used in the tropical tuna (YFT & BET) stock assessments (SA) in the IOTC as one of SA approaches in the WPTT in past 8 years. However, this FORTRAN based ASPM has following limited capabilities:

- (a) Very slow operating speed especially to conduct the bootstrap to estimate variances (it takes more than 10 hours if number of the re-sampling is large);
- (b) It can handle only (max.) 4 fleets;
- (c) It estimates steepness which cannot be fixed that has caused problems in the past, i.e., for many cases in the past steepness are estimated as 0.999 which is unrealistic thus we need to fix the steepness for ASPM runs to get more reasonable results (suggested by late Geoff Kirkwood).

To improve these problems, two ASPM software was newly developed for this time using two platforms, i.e., AD Model Builder and SS-V3. This document describes the outlines of these two software. As these two software are still not fully completed, if someone wants to cite this document please contact the editor first.

2. AD MODEL BUILDER ASPM

Victor Restrepo made the ASPM software using FORTRAN by referring to some pelagic resources assessments by Punt (1992?) in South Africa (ICCAT, 1997). As in that paper Professor Doug Butterworth (University of Cape Town) is involved, we asked him if he could develop the improved ASPM. Then his staff Dr Rebecca Rademeyer, an expert on ASPM was introduced and she re-coded FORTRAN based ASPM using AD Model Builder. Table 1 summarizes differences between FORTAN ASPM and AD Model Builder ASPM. Appendix A provides the User's Guide (draft).

Table 1 Comparison between FORTAN ASPM & AD Model Builder ASPM

	FORATN ASPM	AD Model Builder ASPM
Fleets	4	No limits
steepness	Estimated	Estimated or Fixed between 0 and 1
computing speed	1	5-10 times faster
catch equations	Pope's approximation	Baranov catch equations
minimization routines	simplex algorithm	automatic differentiation
uncertainty (variances) estimation	bootstrap	delta method MCMC
Others		Negative log likelihood is computed without the constants.

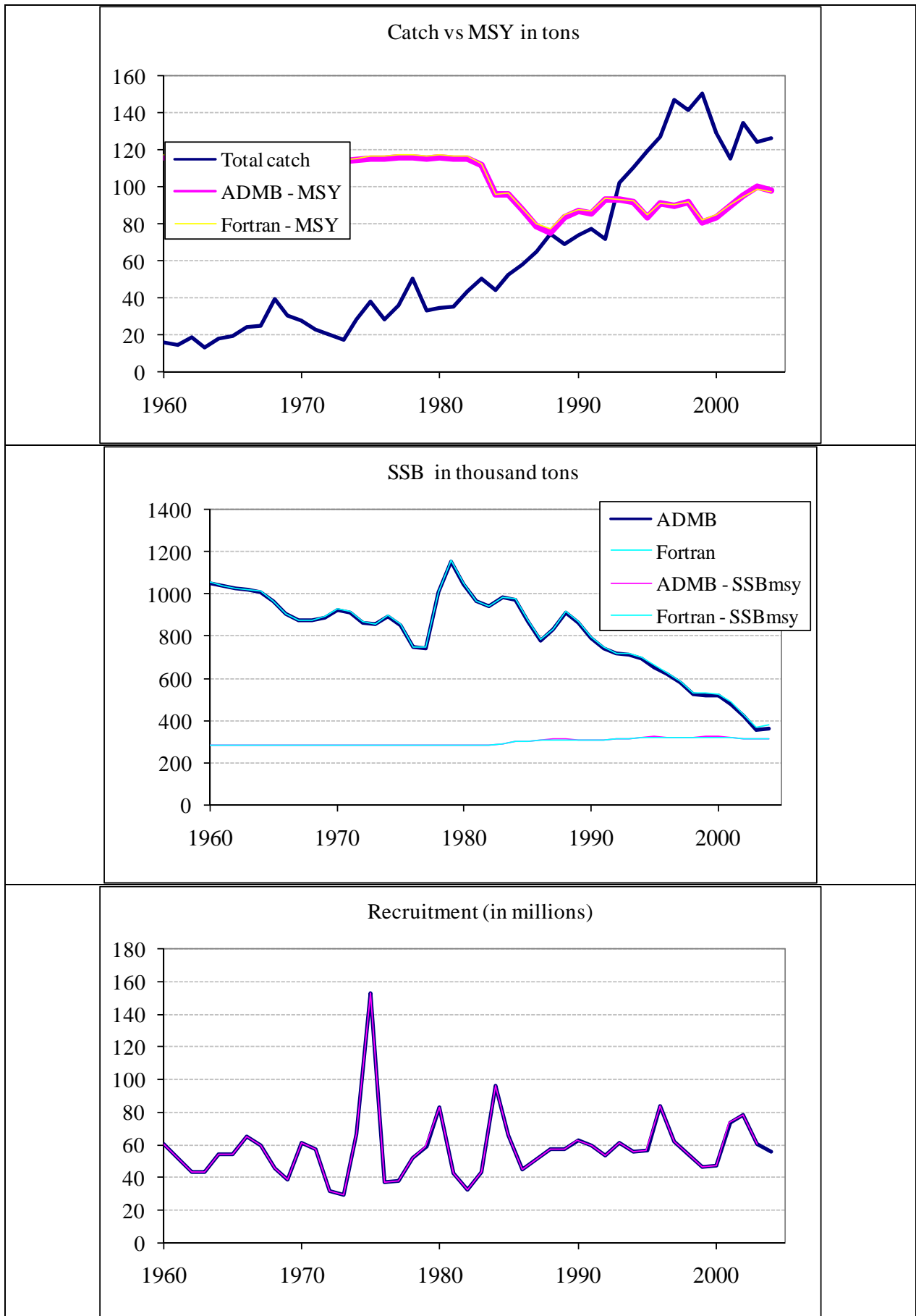
Performance tests

With same bigeye tuna data used in the SA in the WPTT8 in 2006, we compared the SA results between FORTRAN ASPM and ADMB ASPM. The SA results by FORTRAN ASPM were accepted by the WPTT8 and have included in the IOTC's BET Executive Summary.

Fig. 1 shows the results of the performance tests. The results are almost identical. We consider that negligible discrepancies were caused by the different type of optimization methods used between two ASPMs. We consider that FORTARN ASPM picked up the quasi local minima while ADMB ASPM picked up the right ones as the AD Model Builder applies more powerful optimization method. In addition CSIRO scientists in Australia evaluated the FORTAN ASPM and it was concluded that results by the FORTAN ASPM were robust (2004?). Based on these two facts SA results by the FORTRAN ASPM in the past IOTC-WPTTs are valid.

Release of the software

AD Model Builder ASPM software will be released in Public after completed and after approved by the developer.



2. SS3 ASPM

Richard Methot (NOAA, USA) developed SS-V3 software which core structure is similar to the one of the ASPM. For this time he converted SS3 to make it possible to run ASPM (named as SS3 ASPM). This latest SS3 is completed recently and its documentation is still under preparation.

The input files in SS3 ASPM contain many comments to help users to interpret how to make changes. The input files show commented-out (with a #) place-holder values for many inputs that are not being used. Followings are steps how to use the SS3 ASPM.

(1) Translate the ASPM input into the SS3 format

In the spreadsheet ASPM_input.xls there is a page that finds the SS3 growth parameters that will match the ASPM wt-at-age. There also is a page that converts the ASPM selectivity into SS selectivity parameters.

(2) Prepare SS3 input files

4 files are needed, i.e., STARTER.SS, FORECAST.SS, likeASPM.ctl and likeASPM.dat (to arrange 6 fleets)

(3) CPUE index

It is interpreted as one continuous time series because in the likeASPM.ctl file, the section on catchability instructs the model to use the catchability for fleet 1 as the catch-ability for fleets 2 and 3, so on. When SS3 estimates the value for this fleet 1 catchability coefficient, it gets a value that includes data from fleets 2 and 3 also.

(4) Run SS3 ASPM

Type SS3 at a DOS command prompt in the directory containing the model and the input files. Results will be saved in REPORT.SSO file.

Release of the software

SS3 ASPM software has applied only for the test data and not yet applied to the full scale SA data like we did for AD Model Builder ASPM for this time for the YFT SA in WPTT10, 2008. Until real applications and careful examinations are completed and until approvals by the developer are obtained, this software will not be released.

Appendix A

AD Model Builder Implementation of ASPM - Age-structured production model

Users Guide (draft)

Rebecca Rademeyer

IMPLEMENTATION DIFFERENCES

The differences in implementation between the Restrepo Fortran software (ASPM97/ASPMS) and the AD Model Builder (ADMB) software are:

1. The use of the Pope's approximation for the catch equations in the ADMB software (ASPM97/ASPMS uses the Baranov catch equations)
2. Different minimization routines (ASPM97/ASPMS uses a simplex algorithm while ADMB uses automatic differentiation)
3. The ADMB software uses the delta method to derive uncertainty estimates (ASPM97/ASPMS uses bootstrapping)
4. The negative loglikelihood is computed without the constants.

1. HOW TO RUN ASPM

To run ASPM, you need six input files (see next section). One of these is the *control* file, which controls some aspects of the run such as problem dimensions.

To run aspm, type

```
aspm
```

The results will be sent to a series of output files which are described below.

To run with starting values other than those in the *parameter guesses* file (ASPM.pin, see below) (for example in a file named *otherInitial.par*), type

```
aspm -ainp otherInitial.par
```

2. INPUT FILES

Six input files are required, their names cannot be changed: A *control* file, a *parameter guesses*

file, a *biological data* file, an *index* file, a *fishery* file, and a *projection* file. In all these files, “#” precedes a comment line. In all the input files, extra comment lines can be added without affecting the run, as long as these line starts with “#”. The example files contain annotations for input help. Their contents are as follows:

2.1 Control File (named: control.inp)

Line	Entry
1-3	Comments
4	Phase for dummy parameter. Always -1 except if running with ALL the parameters fixed, in that case it needs to be a positive number.
5	Comments
6	First year of catch data
7	Last year of catch data
8	Comment
9	Number of fisheries or gear types for which yield data are available
10	Comment
11	Number of ages (last one is a plus group)
12	Comment
13	Stock-recruitment curve (1= Beverton-Holt, other = Ricker).
14	Comment
15	Phase for estimation of virgin biomass, i.e. whether virgin biomass is estimated (>0) or fixed (to value read in aspm.pin) (-1)
16	Comment
17	Phase for estimation of Binit, i.e. whether Binit is estimated (>0) or fixed (to value read in aspm.pin) (-1)
18	Comment
19	Phase for estimation of steepness, i.e. whether steepness is estimated (>0) or fixed (to value read in aspm.pin) (-1)
20	Comment
21	Whether recruitment is deterministic (-1) or stochastic (>0)
22	Comment
23	Comment
24	Phase for estimation of recruitment deviation. If stochastic recruitment is chosen (line 17 > 0), but want to fix the recruitment deviations in certain years, input -1 for that year
25	Comment
26	Number of indices of relative abundance
27	Comment
28	Objective function type: 1 = lognormal errors, 2 = normal errors.
29	Comment

- 30 Index weighting: 1 = maximum likelihood, 2 = equal weights, -1 = inverse-variance weighting for each point.
- 31 Comment
- 32 B0 multiplier
- 33 Comment
- 34 rho, the lag-1 autocorrelation
- 35 Comment
- 36 sigma squared for the stock-recruitment deviations
- 37 Comment
- 38 sigma squared for B0 fit
- 39 Comment
- 40 weight for the stock-recruitment likelihood component
- 41 Comment
- 42 weight for the B0 fits

Note: The inputs corresponding to “phases” (lines 15, 17, 19 and 24) allow the minimization to be carried out over a subset of the parameters, while the others are fixed. In a non-linear models it can be useful to estimate the different parameters during different phase.

2.2 Parameter guesses File (named: aspm.pin)

- | Line | Entry |
|------|------------------------------------|
| 1 | Comment |
| 2 | Guess for virgin biomass B0 |
| 3 | Comment |
| 4 | Guess for steepness |
| 5-94 | Guesses for recruitment deviations |

Note: If you want the biomass at the beginning of the first year to be the virgin biomass, line 6 must be zero and the first number in line 20 of the control.inp file must be negative (i.e. fix the recruitment deviation to zero for the first year).

2.3 Biological Data File (named: biological.inp)

- | Line | Entry |
|------|--|
| 1-5 | Comments |
| 6 | Weights at age at the start of the year for all ages |
| 7 | Comments |
| 8 | Weights at age at the middle of the year |
| 9 | Comments |
| 10 | Natural mortality at age |

- 11 Comments
- 12 Fecundity at age

2.4 Index File (named: Index.inp)

- | Line | Entry |
|------|--|
| 1-3 | Comments |
| 4 | Index type for each index (1 = biomass, other = numbers). |
| 5 | Comments |
| 6 | Selectivity type for each index: Use straight or scale by F (partial catches). |
| 7 | Comments |
| 8 | Index timing: Month of the year or -1 for mid-year. |
| 9-10 | Comments |

[Repeat this block for each index, can add comments by starting line with #]

11..Nyears: Year, index value (0 for missing), CV(index), selectivity of the index at age.

NOTE: The CV(index) values will be ignored if ML or equal weighting options are selected in the control file.

2.5 Fishery File (named: fishery.inp)

- | Line | Entry |
|----------|---|
| 1-4 | Comments |
| 5-nyears | Year, catch by each fleet for that year |
| ... -... | Annual selectivities at age (at least one = 1) for each fleet |

2.6 Projections File (named: projection.inp)

- | Line | Entry |
|-----------|--|
| 1-3 | Comments |
| 4 | Number of years to project forward |
| 5-6 | Comments |
| 7 | Future catch (constant over the projected period) for each fleet |
| 8-9 | Comments |
| 10-Nfleet | Selectivity-at-age for each fleet |

3. OUTPUT FILES

ASPM creates a series of output files, most of them are standard ADMB output files.

1. ASPM.rep: Results for the run, including spawning biomass, numbers-at-age, recruitment, fits to the data, fishing mortality, MSY and related quantities, etc.
2. proj.out: Projected spawning biomass and projected SSB_{MSY} .
3. ASPM.par: a standard ADMB output file, giving the objective function value, its gradient

(this should be very small if the model has converged) and the parameters estimated/fixed for that run.

4. ASPM.std: a standard ADMB output file, with the parameters estimated for that run and their estimated Hessian-based standard deviation. Annual SSB and their estimated standard deviations are also included.

5. Another useful ADMB standard output files is ASPM.cor for the correlations of the parameter estimates.

4. SUGGESTIONS

- Always try different starting guesses to see if the program converges on the same solution.
- If the model does not converge, you can try restarting it with the parameter that have just been estimated, to do this type:

ASPM -ainp aspm.par

i.e. it will use the values in aspm.par as initial guesses.

- Since the input and output file names are fixed, it's easier to keep each run in a separate folder.