Suggested Modifications to SPM

Started by Sophie Mormede March 2012

The items below are suggested modifications to the SPM code to allow further flexibility and uses.

# BIOMASS

The relevant formulas are described in the SPM manual, sections 4.10 and 4.11. This requires a size at age function, and a size-weight function. Both these functions will be combined in a single age\_size\_weight function, and SPM fuctions will be get\_length and get\_biomass. Each block has only one set of parameters, to be applied to all the categories mentioned; multiple blocks are allowed (say females and males or different species).

Biomass and biomass density layers need to be implemented using these functions (section 4.4). Spawning stock biomass derived quantity also needs to be applied; it is the sum of the spawning biomass layer.

@age\_size\_weight TOA

type von\_bertalanffy

categories immature mature spawning

distribution normal

by\_length True

k 0.093

t0 -0.256

Linf 169.07

Cv 0.10

a 1.387e-008

b 2.965

In the report section, additions need to be made compared with the manual. The output will be for every age from age\_min to age\_max have age, mean weight, mean size, and user-defined quantiles of sizes. Also optionally, the user can define specific lengths at which the corresponding weight will be returned.

@report[label].type=age\_size\_weight

age\_size\_weight defines the label of the age\_size\_weight relationship print

sizes defines the values of the sizes on which to print the weights (optional)

quantiles defines the quantiles on which to report size at age (optional)

# Recruitment

Details of the Beverton Holt (BH) recruitment formula are in the SPM manual, section 4.7.1. The recruitment process needs to define also steepness and YCS for a BH type recruitment.

Two options are defined:

* BH\_global: recruitment in each cell is on the basis of the total recruitment over a large number of cells (an SSB derived quantity over a specific area) and applied to some cells according to a layer (e.g. a recruitment layer), or
* BH\_local: recruitment in each cell is on the basis of the spawning stock biomass of that individual cell only (a biomass layer); note that this recruitment layer can be further submitted to dispersion through a movement function.

@process Recruitment

type BH

categories immature #category to apply the recruitment to

SSB SSB\_TOA # a derived quantity for BH\_global or layer for BH\_local

proportions 1.0

R0 5000000

steepness 0.75

sigma\_r 0.6

standardise\_YCS True

YCS\_years 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009

YCS 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

first\_free 1994

last\_free 2004

So a new type of recruitment needs to be defined, as BH. R0, steepness and sigma can be estimated or fixed, and all the “free” YCS are estimated. If standardise YCS is true, YCS used in the formula is YCS/mean(sum(YCS)), otherwise they are used as defined.

As for constant recruitment, offset is defined as min\_age in the model. Also for initialisation steps, where there is no information on SSB (year 1 to year min\_age), use R0.

# Layers

At the moment if a layer is not the correct size (number of rows and columns), SPM fails, the DOS window closes and a Windows error pops up. When layers get loaded, there is a need to check that they are the correct shape (i.e. same size as the base layer) and give an appropriate error message and exit from SPM.

# preference function for Categorical layers

Currently only numeric layers can be used as covariate layers. The proposed extension would allow categorical layers as covariate layers. The associated preference function would be all values, where all the parameters are estimable. There must be a check whereby the number of categories is the same as the number of values provided, and the categories names match the categories in the layer. There is no provision to specify alpha for this preference function as it would equate to over-parametrisation (it just multiplies the values by an arbitrary factor).

There is provision to force the values to be monotonic, increasing or decreasing. One way of coding the monotonic change is to estimate the first parameter and the difference between each other parameter in order, with those forced to be positive (or negative). In case of monotonic relationship, the categories have to be provided in increasing order.

Type categorical or categorical monotonic (decreasing would be reverting the categories)

f(x) = Vx

@preference\_function ImmatureArea

type categorical

categories a b c

values 10 2 50

# Derived layers

There is a need for more generic derived layers, with specific calculations to obtain these layers, and parameters which can be estimated if needs be. Current derived layers include abundance layers or biomass layers (not yet implemented). A single derived layer module should be able to calculate pre-defined layers such as biomass, and user defined layers. Abundance layer is the sum of all individuals in each cell; biomass layer will be the sum of all individual weights in each cell.

So a new layer type needs to be created, a used-defined calculated layer (see section 8.6). It will include the layers to use, calculation etc. Parameters for the formula can come from other layers, other categories, and / or specifically defined parameters. These parameters can be then estimated in the estimation section as any other parameter. Layer calculations can then be nested in each other.

# example for diet electivity, assuming B\_WGR and B\_ELC are biomass layers

@layer MortalityDep

type user\_defined

parameters ElecTW ElecTE

values 0.75 0.25

formula ElectTW\*B\_WGR/(B\_WGR+B\_ELC)

# then in the estimate section

@estimate layer[MortalityDep].ElecTW

lower\_bound 0.01

upper\_bound 0.99

prior Uniform

# meta layers

These are defined in the SPM manual (section 4.4 item 8). Meta layers are layers indexed by year and applied by year. The meta-layer class is already defined (see section 8.6.8), which is in effect a three dimensional layer. Need to add the type of layer (numeric or categorical).

From the description it seems like the meta-layer is defined as a list of the names of the individual layers, see example below. It might be useful to have another option to define the meta-layer as defining layer, list the data as a layer, with data lines for each year.

Also required is how to apply these to calculations. Everywhere a layer is used, if it is a meta-layer then the year-specific layer should be retrieved prior to being applied.

Interpolation of layers is not allowed: all years defined must be contiguous otherwise throw an error. Extrapolation method is requested in the arguments, whereby the user defines the number of years to average over, or a user-defined layer to apply:

* 0 means all preceding (or following) layers are set at 0 (for example no fishing prior to the start of the fishery)
* any other number is the number of years to average over, and has to be limited to the maximum number of years available. For example 5 would mean the extrapolated value in the future (or past) is the average of the last 5 years. An exception must be thrown if there are less years available than requested (e.g. value 100 when there are 10 years of data).

**Note**: Fishing layers (for example) can be defined as meta-layers or as individual layers. We need to make sure the code can cope with either, whereby the layer is applied, or if a meta layer, the appropriate year is applied. This will impact the @process function.

# example as it is currently described

@layer Fishing

initialisation 0

prediction 5

years 1998 1999

layers Fishing\_1998 Fishing\_1999

# example as it could be described as a second option

@layer Fishing

initialisation 0

prediction FutureFishing

Years 1998 1999

#1998

data 0 0 234 0 111 0

data 0 0 0 500 10 0

… #as many rows as rows in the model

#1999

data 0 0 400 0 80 0

data 0 20 600 0 0

…

# Mortality as an annual rate

Mortality as a constant rate is currently implemented (section 4.7.3), either as a constant, or as a function of a layer. Mortality as an annual rate should be implemented using the same function of a layer, whereby meta-layers can be used, or calculated layers (expected to change every year). If annual, mortality then needs to be calculated every year.

Mortality as a biomass event also needs to be implemented once the biomass part of the model has been implemented. Once again it is all described in SPM.

Two other mortalities need to be defined, as proportions, using the same module but M is used directly and not as exp(-M).

{Sophie to describe in the manual how to use this as a density-dependent mortality}

# Tags as indicators of movement only

The aim of this new implementation is to use tags as informative of movement only, based on individual tag recaptures only. A new observation data class is created which relates to individual event matching. Data is input with a line per recapture event, with the relevant information to the model (in the order specified).

@observation Tag\_match

type event\_match

categories immature\_tag mature\_tag spawning\_tag

# year\_release year\_recapture cell\_release cell\_recapture age\_recapture selectivity\_recapture selectivity\_recapture selectivity\_recapture

data 2004 2009 r3-c4 r3-c4 12 FishingSel FishingSel FishingSel

…

Each tag recapture will be associated with temporary layers, created in the year of release. The layers will be calculated as such

* At year\_release,
  + Estimate the proportion of that fish in each category
  + Create one layer per category with in cell\_release the value of the proportion of that fish in that category
  + Age of that fish is age\_recapture + year\_recapture - year\_release.
* Between year\_release +1 and year\_recapture:
  + For each category, apply the relevant processes in the model (e.g. maturation, movement, mortality)
  + Update the layers to the new values of distribution, moving fish between layers (categories) and cells
* At year of recapture, once the layer has been updated
  + Apply fishing selectivity to each layer
  + For each layer, calculate the probability of being in the recapture\_cell as the value in that cell divided by the sum of the value in all the cells. Add each value to the total likelihood
  + Delete the layers

# Report state

Type: partition, add years instead of year so a single report can give all the years.