# Age-structured model for Alaska herring stocks

Steve Martell

July 25, 2016

**Abstract**

# 1 Introduction

# 2 Model deconstruction

This section is intended to serve three purposes: 1) to document the model structure given the code in model.tpl, 2) to detail proposed changes to the model code to improve overall numerical stability, and 3) provide statistical approach that is amenable to maximum likelihood and Bayesian methods.

## 2.1 Model Structure

Table 1 begins with the objective function that is being minimized. There are four components defined in (T1.1), where three of the four components are scaled by user defined coefficients $\lambda$. The first component is the commercial age-composition data ($QC$), the second is the spawnig biomass age-composition ($QS$), the third is egg deposition data ($WQE$), and finally the fourth component is a penalty on the recruitment deviations from the underlying Ricker stock-recruitment model ($QR$).

For the commercial age-composition data, observation errors in the age-proportions are assumed to be normal (T1.2), where the predicted proportion-at-age (T1.3) is a function of the numbers-at-age (T1.4) and selectivity (T1.5). Note that in (T1.4) that the function is not continous. In this case the selectivity curve is rescaled to have a maximum value of 1.0. The `max` operation in the denominator of this function breaks the derivative chain in AD Model Builder and can result in numerical problems during parameter optimization associated with corrupt derivative information.

The same normal error distribution is also assumed for the age-proportions in the spawner catch composition samples (T1.6). In this case, the age proportions aree based on the mature numbers-at-age at the time of spawning, where the fisehries removals are first subtracted from the mature numbers-at-age (T1.7). Note that this further assumes that all removals (i.e.,

fisheries selectivity) only harvests sexually mature fish. This assumption is inconsistent with (T1.4), where a different logistic curve is assumed for fisheries selectivity.

The catch-at-age data is internally derived in the model (T1.9) conditional on the numbers-at-age and the estimated selectivity. The model further assumes the total catch (in short tons) is measured without error. This is also referred to as conditioning the model on catch.

The residual sum of squares for the egg deposition suvey is given by (T1.10). In this case the obsrevation errors are assumed to be lognormal, and each year's observation is weighted by the inverse variance of sampling observation errors ($\varphi_i$).

Table 1: Decomposition of the objective function based on the source code provided in `model.tpl`. The objective function $f$ is what AD Model Builder is trying to minimize. Note that $\dot{}$ represents mature state variables (e.g., mature weight-at-age $\dot{w}_j$)

Objective function

$$f = \lambda_C QC + \lambda_S QS + WQE + \lambda_R QR \tag{T1.1}$$

Commercial catch proportion-at-age

$$QC = \sum_i \sum_j (\hat{Q}_{i,j} - Q_{i,j})^2 \qquad \hat{Q}_{i,j} \text{ observed proportions-at-age} \tag{T1.2}$$

$$Q_{i,j} = \frac{V_{i,j}}{\sum_j V_{i,j}} \qquad Q_{i,j} \text{ predicted proportion-at-age} \tag{T1.3}$$

$$V_{i,j} = N_{i,j} \frac{S_{i,j}}{\max(S_{i,j})} \qquad V_{i,j} \text{ vulnerable numbers-at-age} \tag{T1.4}$$

$$S_{i,j} = \frac{1}{1 + \exp(-g_i(j - a_i))} \qquad S_{i,j} \text{ Selectivity in year } i \text{ for age } j \tag{T1.5}$$

Spawn proportion-at-age

$$QS = \sum_i \sum_j (\hat{O}_{i,j} - O_{i,j})^2 \qquad \hat{O}_{i,j} \text{ observed proportion-at-age} \tag{T1.6}$$

$$O_{i,j} = \frac{\dot{N}_{i,j}}{\sum_j \dot{N}_{i,j}} \qquad O_{i,j} \text{ predicted proportion mature-at-age} \tag{T1.7}$$

$$\dot{N}_{i,j} = N_{i,j} M_{i,j} - C_{i,j} \qquad \dot{N}_{ij} \text{ Number-at-age at spawning} \tag{T1.8}$$

$$C_{i,j} = \frac{\hat{c}_i Q_{i,j}}{\sum_j Q_{i,j} w_j} \qquad \hat{c}_i \text{ observed catch, } w_j \text{ weight-at-age} \tag{T1.9}$$

Egg deposition survey

$$WQE = \sum_i \varphi_i \left[\ln(\hat{y}_i) - \ln(y_i)\right]^2 \qquad \hat{y}_i \text{ observed egg deposition, } \varphi_i \text{ weight} \tag{T1.10}$$

$$y_i = 0.5 \sum_j O_{i,j}(\rho_i \dot{w}_{i,j} - \alpha_i) \qquad \rho_i \text{ and } \alpha_i \text{ fecundity-weight regression} \tag{T1.11}$$

Penalized recruitment deviations

$$QR = \sum_i^{(I-k)} \left[\ln\left(N_{i+k,k}\right) - \ln\left(f(\dot{N}_{i,j})\right)\right]^2 \qquad N_{i+k,k} \text{ number of age } k \text{ recruits} \tag{T1.12}$$

$$f(\dot{N}_{i,j}) = a\dot{B}_i \exp(-b\dot{B}_i) \qquad \text{Ricker stock-recruitment} \tag{T1.13}$$

$$\dot{B}_i = \sum_j \dot{N}_{i,j} \dot{w}_j \qquad \dot{B}_i \text{ mature biomass at spawning} \tag{T1.14}$$

# 3 Technical description of the proposed model changes

## 3.1 Input Data

The best resourcee for looking at the input data is the html file that describes the input data. There are 7 major sections to the data file.

1. Model dimensions.
2. Fecundity regression coefficients.
3. Total Annual Catch.
4. Empirical Weight-at-age (spawn and commercial).
5. Age-composition (spawn and commercial).
6. Egg deposition data.
7. Mile milt days.

These are the same data inputs that were used in the ASA model; however, there have a been a number of siginificant changes to the input data file. The most siginificant change is the addition of the log standar error for each catch, egg deposition, and spawn survey observation.

## 3.2 Control file

There are also siginficant changes to the control file. In fact, its a completely different control file than what was used in the ASA model. Again, the best resource for looking at the specific contents of the control file, is the control file itself.

To highlight some of the major changes, the control file now consists of a design matrix for controling the leading model parameters; specificially, the bounds and phases in which these parameters are estimated. There is a block for time-varying maturity, a block for time-varying naural mortality rate deviations. A block for selectivity, where the user can choose among alternative parametric and non-parametric selectivity curves. Lastly, there is a vector of other miscellaneous model controls for, *inter alia*, re-scaling catch, conditioning the model on catch or effort.

## 3.3 Age-schedule information

Age-schedule information is part date input: weight-at-age, or maturity-at-age via regression coefficients. The other part of age-schedule information is estimated based on fitting the model to data: selectivity-at-age, maturity-at-age.

### 3.3.1 Maturity-at-age

For the maturity-at-age, the HAM assumes that age-specific maturity follows a logistic relationship, where the estimated paraemeters define the age-at-50% maturity and the standard deviation, for each unique block period (the block periods are user defined).

The source code for the TPL file is as follows:

```
FUNCTION void initializeMaturitySchedules()
  int iyr = mod_syr;
  int jyr;
  mat.initialize();
  for(int h = 1; h <= nMatBlocks; h++) {
    dvariable mat_a = mat_params(h,1);
    dvariable mat_b = mat_params(h,2);

    jyr = h != nMatBlocks ? nMatBlockYear(h) : nMatBlockYear(h)-retro_yrs;
    // fill maturity array using logistic function
    do{
      mat(iyr++) = plogis(age,mat_a,mat_b);
    } while(iyr <= jyr);
  }
```

where `plogis` is a built in ADMB function for the logistic:

$$f(x) = (1 + \exp(-(x - \mu)/\sigma))^{-1}$$

where $\mu$ and $\sigma$ are the location and scale parameters that are estimated.

### 3.3.2 Natural mortality

Natural mortality is both age- and time-specific. At the time of writing, there is no code that allows for age-dependent natural mortality, but this option is easily added as a feature to the HAM.

The source code for the TPL file is as follows:

```
FUNCTION void calcNaturalMortality()

  int iyr = mod_syr;
  int jyr;
  Mij.initialize();

  for(int h = 1; h <= nMortBlocks; h++){
    dvariable mi = mfexp(log_natural_mortality + log_m_devs(h));

    jyr = h != nMortBlocks ? nMortBlockYear(h) : nMortBlockYear(h)-retro_yrs;
    // fill mortality array by block
```

```
    do{
      Mij(iyr++) = mi;
    } while(iyr <= jyr);
  }
  //COUT(Mij)
```

where the Matrix $M_{i,j}$ is the instananeous natural mortality rate for year $i$ and age $j$. At this point the code just fills each row of the matrix with the same annual natural mortality rate (i.e., age-independent). This is where you would want to modify the code to allow for age-dependent natural mortality rates.

### 3.3.3 Selectivity

Currently only the logistic selectivity option is implemented. But the source code is structured such that alternative parametric and non-parametric functions can be easily added to the source code using a swtich statement.

The source code for the TPL file is as follows:

```
FUNCTION void calcSelectivity()
  /**
    - Loop over each of the selectivity block/pattern
    - Determine which selectivity type is being used.
    - get parameters from log_slx_pars
    - calculate the age-dependent selectivity pattern
    - fill selectivty array for that block.
    - selectivity is scaled to have a mean = 1 across all ages.
  */
  dvariable p1,p2;
  dvar_vector slx(sage,nage);
  log_slx.initialize();



  for(int h = 1; h <= nSlxBlks; h++){

    switch(nSelType(h)){
      case 1: //logistic
        p1  = mfexp(log_slx_pars(h,1,1));
        p2  = mfexp(log_slx_pars(h,1,2));
        slx = plogis(age,p1,p2) + TINY;
      break;
    }

    int jyr = h != nSlxBlks ? nslx_nyr(h):nslx_nyr(h)-retro_yrs;
```

```
    for(int i = nslx_syr(h); i <= jyr; i++){
      log_slx(i) = log(slx) - log(mean(slx));
    }
  }
  Sij.sub(mod_syr,mod_nyr) = mfexp(log_slx);
```

The matrix $S_{i,j}$ is the relative selectivity for age $j$ in year $i$. Additional functions for computing the vector `slx` can be new cases (e.g. case 2: // coefficients).

There is a very important feature in the selectivity is parameterized in this model. The reason for this particular parameterization is to ensure the objective function remains continuous and differentiable. In each year, the vector of age-specific selectivity coefficients is scaled to have a mean value of 1.0. This is accomplished, in log-space, but subtracting the mean from the vector of age-specific selectivities. This avoids having to use the max function; the use of the max function can lead to a discontinuity in the objective function which result in non-convergence. The tradeoff for this numerical stability is that the interpretation of Fishing mortality rates changes. The estimator is calculating the average-age-specific fishing mortality rate. The fully-selected fishing mortality rate is more commonly used metric, and this is easily accommodated post-estimation by re-scaling the vector of fishing mortality rates by the maximum age-specific selectivity in each year.

If the previous paragraph didn't make sense, go read the R-code.

## 3.4 Fishing mortality

If the model is conditioned on effort, then a routine that calculates the age-specific fishing mortality rate is invoked. In this routine, a vector of fishing mortality rate parametes, in log-space, is estimated, and the age-specific fishing mortality rate is the product of the fishing rate and age-specific selectivity. The source code for this routine is as follows:

```
FUNCTION void calcFishingMortalitiy()
  /**
    - Calculate Fishing mortality, and then Zij = Mij + Fij
    */

  for(int i = mod_syr; i <= mod_nyr; i++) {
    Fij(i) = exp(log_ft_pars(i)) * Sij(i);
  }
```

If the model is conditioned on catch (NB this is the method the ASA model uses), then there is a resulting difference equation which has the potential to result in negative numbers-at-age, which results in negative Infinity in log-space. To guard against this, many excel users would simply use a Max function to ensure that a positive number is returned. This is another occurance where the objective function is disconituous and subject to non-convergence issues. AD Model Builder has a function `posfun` that can be used to ensure the objective function remains continuous and differentiable.

## 3.5 Population dynamics

Estimated parameters for the population dynamics model include the initial abundance of ages 3-9+ for the intial year, abundance of age-3 recruits each year, and the natural mortality rate. In the original parameterizeation of the model, these initial recruitments and the vector of initial numbers-at-age result in creating $(N + A - 1)$ scaling parameters. To reduce the potential confounding with other global scaling parmaters, updates to the model code include estimation of two recruitment scaling parameters, and two vectors of deviates that represent deviations from the mean. This modification reduces the potential for parameter confounding among the many paraemters that affect global scaling (i.e., catchabiltiy coefficients, natural mortlaity rates).

Table 2: Notation and equations for population dynamics model.

| | | | |
|---|---|---|---|
| Model parameters | | | |
| $\theta = \{\ln(M), \ln(\bar{R}), \ln(\ddot{R}), \ln(\alpha), \ln(\beta), \vec{\delta}\}$ | | | (T2.1) |
| Initial States ($i = 1980$) | | | |
| $\iota_j = \begin{cases} \exp(-M * (j - \min(j))) & \text{where } 3 \leq j \leq 7 \\ \dfrac{\exp(-M * (j - \min(j)))}{1 - \exp(-M)} & \text{where } j = 8 \end{cases}$ | | survivorship | (T2.2) |
| $N_{i,j} = \ddot{R} \exp(\delta_{i-j}) \iota_j$ | $i = 1980, \forall j$ | initial numbers-at-age | (T2.3) |
| $N_{i,j} = \bar{R} \exp(\delta_i)$ | $\forall i, j = 3$ | age-3 recruits | (T2.4) |
| Dynamic States ($i > 1980$) | | | |
| $Q_{i,j} = \dfrac{N_{i,j} S_{i,j}}{\sum_j N_{i,j} S_{i,j}}$ | | vulnerable proportions | (T2.5) |
| $\bar{w}_i = \sum_j w_j Q_{i,j}$ | | average weight of catch | (T2.6) |
| $C_{i,j} = \dfrac{\hat{c}_i Q_{i,j}}{\bar{w}_i}$ | where $\hat{c}_i$ is the observed catch (mt) | $C_{i,j}$ catch-at-age | (T2.7) |
| $\acute{N}_{i,j} = N_{i,j} \exp(-F_{i,j})$ | | | (T2.8) |
| $N_{i+1,j+1} = \acute{N}_{i,j} \exp(-M_{i,j})$ | | | (T2.9) |

### 3.5.1 Initial state variables.

In this routine, the objective is to set the initial values for the numbers-at-age matrix $N_{i,j}$. Specifically, the row dimensions of the matrix are from the start year to the terminal year

+ 1, the column dimensions are the ages. This routine first calculates the survivorship to age $j$ based on natural mortality rates, then initializes the first row of $N_{i,j}$ matrix using the average initial recruitment and deviations around that average recruitment multiplied by the survivorship at age $j$. The source code also includes the taylor series for the plus group:

```
FUNCTION void initializeStateVariables()
  /**
    - Set initial values for numbers-at-age matrix in first year
      and sage recruits for all years.
    */
  Nij.initialize();

  // initialize first row of numbers-at-age matrix
  // lx is a vector of survivorship (probability of surviving to age j)
  dvar_vector lx(sage,nage);

  for(int j = sage; j <= nage; j++){

    lx(j) = exp(-Mij(mod_syr,j)*(j-sage));
    if( j==nage ) lx(j) /= (1.0-exp(-Mij(mod_syr,j)));


    if( j > sage ){
      Nij(mod_syr)(j) = mfexp(log_rinit + log_rinit_devs(j)) * lx(j);
    }
  }



  // iniitialize first column of numbers-at-age matrix
  for(int i = mod_syr; i <= mod_nyr + 1; i++){
    Nij(i,sage) = mfexp(log_rbar + log_rbar_devs(i));
  }
```

The survivorship calculation `lx` correspondds to (T2.2) in Table 2, and the numbers-at-age in the initial year and age-3 recruits corresponds to (T2.3) and (T2.4), respectively.

### 3.5.2 Update state variables

In this routine, the numbers-at-age are propogated in time, where the-age specific survival rate is partitioned into two periods: a fishing period, and a period of natural mortality. The ASA model currently in use for Sitka sound herring assumes a pulse fishery. At the start of each time step, the model first calculates the predicted catch-at-age in numbers in year $i$. This is done by first converting the catch weight to catch in numbers by dividing by the

predicted average of the catch. This corresponds to the `wbar` variable in the following code chunk (note the dependencey on predicted proportions-at-age):

```
FUNCTION void updateStateVariables()
  /**
    - Update the numbers-at-age conditional on the catch-at-age.
    - Assume a pulse fishery.
    - step 1 calculate a vector of vulnerable-numbers-at-age
    - step 2 calculate vulnerable proportions-at-age.
    - step 3 calc average weight of catch (wbar) conditional on Qij.
    - step 4 calc catch-at-age | catch in biomass Cij = Ct/wbar * Qij.
    - step 5 condition on Ft or else condition on observed catch.
    - step 6 update numbers-at-age (using a very dangerous difference eqn.)
    */

    Qij.initialize();
    Cij.initialize();
    Pij.initialize();
    dvariable wbar;   // average weight of the catch.
    dvar_vector vj(sage,nage);
    //dvar_vector pj(sage,nage);
    dvar_vector sj(sage,nage);


    for(int i = mod_syr; i <= mod_nyr; i++){


      // step 1.
      vj = elem_prod(Nij(i),Sij(i));

      // step 2.
      Qij(i) = vj / sum(vj);

      // ADF&G's approach.
      if( !dMiscCont(2) ) {
        // step 3.
        dvector wa = data_cm_waa(i)(sage,nage);
        wbar = wa * Qij(i);

        // step 4.
        Cij(i) = data_catch(i,2) / wbar * Qij(i);
        Pij(i) = posfun(Nij(i) - Cij(i),0.01,fpen); // should use posfun here
```

```
    // step 6. update numbers at age
    sj = mfexp(-Mij(i));
    Nij(i+1)(sage+1,nage) =++ elem_prod(Pij(i)(sage,nage-1),sj(sage,nage-
    Nij(i+1)(nage) += Pij(i,nage) * sj(nage);
  }
  // step 5.
  // Condition on Ft
  else {
    // add flexibility here for Popes approx, or different seasons
    Pij(i) = elem_prod( Nij(i), exp(-Fij(i)) );
    Cij(i) = elem_prod( Nij(i), 1.-exp(-Fij(i)));

    // the following assumes fishery is at the start of the year.
    dvar_vector zi = Mij(i) + Fij(i);
    Nij(i+1)(sage+1,nage) = ++ elem_prod(Nij(i)(sage,nage-1),
```

Once the catch-at-age data is calculated, the pulse fishery proceeds by subtracting the $C_{ij}$ from the $N_{i,j}$. The last two steps correspond to step 4 in the annotated code chunk. The last steps the survive the remaining numbers-at-age using the natrual mortality rate in year $i$. Then finally, update the numbers-at-age $j$ to $j+1$ in year $i$ to year $i+1$.

# References

Table 3: Mathematical notation, symbols and descriptions.

| Symbol | Description |
|---|---|
| Index | |
| $g$ | group |
| $h$ | sex |
| $i$ | year |
| $j$ | time step (years) |
| $k$ | gear or fleet |
| $l$ | index for length class |
| $m$ | index for maturity state |
| $o$ | index for shell condition. |
| Leading Model Parameters | |
| $M$ | Instantaneous natural mortality rate |
| $\bar{R}$ | Average recruitment |
| $\ddot{R}$ | Initial recruitment |
| $\alpha_r$ | Mode of size-at-recruitment |
| $\beta_r$ | Shape parameter for size-at-recruitment |
| $R_0$ | Unfished average recruitment |
| $\kappa$ | Recruitment compensation ratio |
| Size schedule information | |
| $w_{h,l}$ | Mean weight-at-length $l$ |
| $m_{h,l}$ | Average proportion mature-at-length $l$ |
| Per recruit incidence functions | |
| $\phi_B$ | Spawning biomass per recruit |
| $\phi_{Q_k}$ | Yield per recruit for fishery $k$ |
| $\phi_{Y_k}$ | Retained catch per recruit for fishery $k$ |
| $\phi_{D_k}$ | Discarded catch per recruit for fishery $k$ |
| Selectivity parameters | |
| $a_{h,k,l}$ | Length at 50% selectivity in length interval $l$ |
| $\sigma_{s_{h,k}}$ | Standard deviation in length-at-selectivity |
| $r_{h,k,l}$ | Length at 50% retention |
| $\sigma_{y_{h,k}}$ | Standard deviation in length-at-retention |
| $\xi_{h,k}$ | Discard mortality rate for gear $k$ and sex $h$ |