GeoSAMS

Table of Contents

[1 Initialize simulation parameters 2](#_Toc152919068)

[1.1 Read\_Input 2](#_Toc152919069)

[1.2 Set\_Growth 2](#_Toc152919070)

[1.2.1 Load\_Grid and Initial State 2](#_Toc152919071)

[1.2.2 Define shell\_lengths in mm and conversion to weight in grams, for each class 2](#_Toc152919072)

[1.2.3 Computes Growth Parameters, L∞µ, Kmu, L∞sd, Ksd based on depth, latitude, and if grid is closed 3](#_Toc152919073)

[1.2.4 Computes G matrix for given growth parameters 3](#_Toc152919074)

[1.3 Set Recruitment 4](#_Toc152919075)

[1.3.1 Recruitment data 4](#_Toc152919076)

[1.3.2 This method is effectively setting 4](#_Toc152919077)

[1.3.3 It then quantizes recruitment, 4](#_Toc152919078)

[1.4 Set Mortality 4](#_Toc152919079)

[1.4.1 Decreasing logistic function 5](#_Toc152919080)

[1.4.2 Fishing Effort 5](#_Toc152919081)

[2 Main Loop 5](#_Toc152919082)

[2.1 Determine Selectivity 5](#_Toc152919083)

[2.2 Set Fishing Effort 6](#_Toc152919084)

[2.2.1 Determine Total Catch. Landings 6](#_Toc152919085)

[2.2.2 USD 6](#_Toc152919086)

[2.2.3 Fishing Effort by Weight 7](#_Toc152919087)

[2.3 For each grid 7](#_Toc152919088)

[2.3.1 Computes natural mortality based on current state. 7](#_Toc152919089)

[2.3.2 Adjust population state based on von Bertalanffy growth 7](#_Toc152919090)

[2.3.3 Computes increase in population due to recruitment, 8](#_Toc152919091)

[2.3.4 Compute Overall Mortality 8](#_Toc152919092)

[2.3.5 Compute effect of mortality to arrive at new state 8](#_Toc152919093)

# Initialize simulation parameters

## Read\_Input

For example, the current file is focused on MA

* Domain Name = MA
* Beging Year = 2000
* Ending Year = 2000
* Fishing type = 'USD'
* Time steps per Year = 52
* Initial Conditions [number of grids by number of size classes]

File Name = Data/bin5mm2000MA.csv [number of grids by number of size classes]  
***Note: this data is pulled from dredge survey data in OriginalData/dredgetowbysize7917.csv***

## Set\_Growth

The simulation then instantiates parameters that define how growth occurs

### Load\_Grid and Initial State

The data in each file, Data/bin5mmYYYY[MA|GB].csv has grid information of where each grid is located and its depth. Data in the same row is used for the initial state, in units of scallop count per square for each size classs.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Dec Year | UTM X | UTM Y | Latitude | Longitude | Depth | Is Closed |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 30mm | 35mm | 40mm | 45mm | 50mm | 55mm | 60mm | 65mm | 70mm | 75mm |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 80mm | 85mm | 90mm | 95mm | 100mm | 105mm | 110mm | 115mm | … 150mm |

### Define shell\_lengths in mm and conversion to weight in grams, for each class

#### Shell length is a simple equation

Starting at 30mm to 150mm inclusive, in 5 mm steps.

That is (150 – 30) / 5 + 1, or 25 size classes

#### Shell weight is dependent on grid location and depth

GB

Shell\_to\_Weight = exp( – 6.69 + 2.878 \* log(shell\_length\_mm) – 0.0073 \* depth  
 – 0.073 \* latitude + (1.28 – 0.25 \* log(shell\_length\_mm)) \* Logic\_To\_Double(is\_closed) )

MA

Shell\_to\_Weight = exp( – 9.713394 + 2.62025 \* log(shell\_length\_mm) – 0.004665 \* depth  
+ 0.021 \* latitude – 0.031 \* Logic\_To\_Double(is\_closed))

### Computes Growth Parameters, L∞µ, Kmu, L∞sd, Ksd based on depth, latitude, and isClosed

#### Truncate the larger size bins

And for each grid, the length values that are greater than L∞µ are gathered into the same bin.

### Computes G matrix for given growth parameters

From MN18 p. 1312, 1313

\[
 c = 1.0 - e^{-K_{\mu} * \delta_t}
 \]

\[
 \eta = c * L_{{\infty}_\mu}
 \]

For each size class, k

* \[
   \omega_k = l_k - l_{k-1}
   \]
* \[
   \omega_{k_{avg}} = \frac{l_k + l_{k-1}}{2}
   \]
* Compute a growth increment distribution parameter, **σ**   
  increment\_mean\_std (L∞µ, Kmu, L∞sd, Ksd, ωavg, µ, **σ**)
* \[
   \Omega = (1 - c) \omega_k
   \]
* \[
   X(y,k) = l_y - \eta - (1-c)l_{k}
   \]
* \[
   G(y, k, \sigma, \omega_k) = H_{MN18}(X(y,k-1), \sigma, \Omega)\\
                             - H_{MN18}(X(y,k),   \sigma, \Omega)
   \]
* \[
   H_{MN18}(x, \sigma, \omega) = \frac{1}{\omega}\left[x\Phi_N(x,0,\sigma^2) + \sigma^2\phi_N(x,0,\sigma^2)\right]
   \]

\[
 \Phi(x,\mu,\sigma) = \frac{1}{2}(1+Erf(\frac{x-\mu}{\sigma\sqrt{2}}))
 \]

\[
 \phi(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}
 \]

## Set Recruitment

The simulation next instantiates how recruitment will be handled.

### Recruitment data

#### For years 1979, 2018

Data is read in from KrigingEstimates/Sim[MA|GB]YYYY/KrigingEstimate.txt

#### For years 2019, 2025

Data is read in from KrigingEstimates/Sim[MA|GB]Clim//KrigingEstimate.txt

### This method is effectively setting

For all years

Year\_index = year - 1978

for year\_index in [1..max]

recruitment(year\_index) = KrigingEstimate

year(year\_index) = year

rec\_start = 1/365, or January 1st

rec\_stop = 100/365, or April 10

### It then quantizes recruitment,

For each grid, n

* L30mm = (L∞µ(n) – 30) \* exp(-Kµ(n))
* For each class, j
  + If (length(n) <= L30mm) recruit(n).max\_rec\_ind = j

## Set Mortality

The simulation next instantiates how mortality is defined.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Mortality | | l0 |
| Natural, Adult | Incidental |
| MA | 25% | 5% | 65.0 |
| GB | 20% | 10% | 70.0 |

### Decreasing logistic function

Specified α

\[
 \alpha(length) = 1-\frac{1}{1+e^{-length_0[length-a]}}
 \]

Implemented α

\[
 \alpha(length) = 1-\frac{1}{1+e^{- a*( length/10.0-length_0 )}}
 \]

Value is irrelevant, see section 2.3.1

### Fishing Effort

Finally, fishing effort is defined by year and region

Read in ***Data/FYrGBcGBoMA.csv***

|  |  |  |  |
| --- | --- | --- | --- |
| **Year** | **GB Closed** | **GB Open** | **MA** |
| 2000 | 0.07 | 0.54 | 0.42 |

If domain is MA  
Mortality (#grids).fishingEffort(yr\_index) = ***FYrGBcGBoMA(yr\_index, 4)***

Else  
If grid posn is closed

Mortality (#grids).fishingEffort(yr\_index) = ***FYrGBcGBoMA(yr\_index, 2)***

Else Mortality(#grids).fishingEffort(yr\_index) = ***FYrGBcGBoMA(yr\_index, 3)***

# Main Loop

**For each year**

## Determine Selectivity

Selectivity is a function of shell length and dependent on year and domain. In general, it is computed as

\[
 Selectivity = \frac{1}{ 1 + e^{ a - b * length_{shell}}}
 \]

Where a and b are based on year and domain.

*Discard* is set at 20% of selectivity unless shell size is > 90 mm (or 100mm and is\_closed for GB) for which case ***discard*** is 0.0

## Set Fishing Effort

Here there is defined a fishing effort that is independent of mortality. Whereas the mortality fishing effort is a function of region and historical data, this fishing effort is a function of cost, biomass or as a spatial constant within region. For example, the following is by cost.

### Determine Total Catch. Landings

Read ***Data/Landings\_75-19nh.csv***, years 1975 to 2019

|  |  |  |  |
| --- | --- | --- | --- |
| **Year** | **Total Catch** | | |
| **GB Closed** | **GB Open** | **MA** |
| 2000 | 2346.47 | 2697.53 | 9351 |

For year 2000 and MA, total catch is given as 9531.

Proposed Change:

Landings - these are calculated as the removals due to fishing mortality (not including discards and incidental mortality). Landings, in terms of numbers by size bin and location, can be calculated as:

NumberLandedAtSize =

(1-exp(-F\*timestep))\*PopulationNumberAtSize\*SelectivityAtSize

F: Fishing Mortality, i.e. Fishing Effort?

PopulationNumberAtSize: S \* domain area

SelectivityAtSize: Selectivity

which would be looped through (or better, use elemental vector multiplication) all sizes > CullSize and all spatial locations.

You can then add these up to get total number of scallops landed and, using the shell height to meat weight conversion, weight of landings by location, and then sum over locations to output total landings in terms of numbers and biomass. The landings by location should also be saved somewhere - eventually, we will interpolate this to make maps of landings.

NumberLandedAtSize =

(1-exp(-F\*timestep))\*PopulationNumberAtSize\*SelectivityAtSize

PopulationNumberAtSize\*SelectivityAtSize   
population = mortality(j)%select(1:num\_size\_classes)  
 \* state(j, 1:num\_size\_classes) \* grid\_area\_sqm

Fishing Effort is a function of Data/Landings\_75-19nh.csv used to determine TotalCatch   
It would become a cyclic computation

### USD

#### Determine viable scallops per square meter

scallops\_per\_sqm(1:num\_grids) = selectivity \* state(1:num\_grids)

#### Convert weight in grams to pounds per scallop and sort by weight

* > 0.1 lbs into cnt10
* <= 0.1 and > 0.02 lbs into cnt20
* <= 0.2 and > 0.03333 lbs into cnt30
* <= 0.03333 into cnt30plus

#### Scallop Price

Data is read from ***Data/ScallopPrice.csv***, years 1998 to 2021

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Year** | **Cnt10** | **Cnt10-20** | **Cnt20-30** | **Cnt30+** |
| 2000 | 6.8548 | 5.30293 | 4.66137 | 5.05687 |

USD\_per\_pound = cnt10 \* scallopPrice(1)  
 + cnt10to20 \* scallopPrice(2)  
 + cnt20to30 \* scallopPrice(3)  
 + cnt30plus \* scallopPrice(4)

* Determine total weight of scallops in pounds  
  totalWeight\_lbs(1:num\_grids) = sum(scallops\_per\_sqm(1:num\_grids) \* weight\_grams(1:num\_grids) / grams\_per\_pound)
* Compute worth in dollars  
  TotalDollars(1:num\_grids) = USD\_per\_pound(1:num\_grids) \* grid\_area\_sqm \* totalWeight \_lbs(1:num\_grids)

### Fishing Effort by Weight

* Fishing Effort by Weight in USD(1:num\_grids) = (TotalCatch / TotalDollars(1:num\_grids)) \* USD\_per\_pound(1:num\_grids)

NOTE: This is the same as

(TotalCatch / { USD\_per\_pound(:) \* grid\_area\_sqm \* totalWeight \_lbs}) \* USD\_per\_pound(:)

USD\_per\_pound cancels out and you are left with (not exactly, but true to x.xxx Esyy)

(TotalCatch / grid\_area\_sqm \* totalWeight \_lbs)

## For each grid

At each time step, δt

### Computes natural mortality

Determine the number of scallops in millions, S, given the current state

S = state \* domainArea

This is used to determine the juvenile mortality.

Mid-Atlantic

\[
 M_{juv} = \begin{cases} 
         e^{1.093 * log(S) - 9.701}, & \text{if } S > 1400 \text{ million (2030?)} \\
         M_{adult},                                   & \text{otherwise}
 \end{cases}
 \]

Georges Bank

\[
 M_{juv} = \begin{cases} 
         e^{(1.226*log(S)-10.49)}, &  \text{if } S > 1400 \text{ million (2030?)} \\
         M_{adult},                                  & \text{otherwise}
 \end{cases}
 \]

where Madult is 0.25 for MA or 0.2 for GB.

At present the computation does not use the conditional but rather whichever is greater, which turns out to be Madult

\[
 M_{nat} = \alpha * M_{juv} + (1-\alpha) M_{adult}
 \]

Further, S starts out at approx. 200 million. To exceed 0.25, we would need S >= 2012.7012 million. Since we have a much, much smaller domain area, num\_grids << 11631, it is not likely we will exceed this value so Mjuv will always be equal to Madult. Thus α is not truly used.

Mnat = α \* Madult + (1- α)Madult

= (α + 1- α)Madult

Mnat = Madult

### Adjust population state based on von Bertalanffy growth

\[
 \vec{S} = \left| G \right| \times \vec{S} 
 \]

### Computes increase in population due to recruitment,

If within recruitment period, i.e. Jan to April 10th

\[
 \vec{S} = \vec{S} + \delta_t\frac{\vec{R}}{RecruitDuration}
 \]

### Compute Overall Mortality

### Compute effect of mortality to arrive at new state

\[
 \vec{S_{t+1}} = \vec{S_t} * (1- \delta_t * \vec{M})
 \]