So ICM takes a few fishery related inputs and several life history traits and then uses some simple relationship to take get an estimate of the population abundance over time and can project out into the future.

**How I think it works** *(This is a walk through of the function por.sim.r from point 2 onwards)*

1. Get an estimate of removals over time from the fishery (this is in the N.options object at the moment), this is in numbers (see ICM\_runme.R)
2. We then use life history parameters and their variability to do a backwards simulation to reconstruct possible population trajectories that got us to where we are.
3. Get a number of life history parameters for your stocks/species using the parms.calc() function, these include
   1. *Von B relationship*: parameters, , *k,*  and the standard errors of these
   2. *Maturity ogive*: Logistic model that needs maturity parameters slope and intercept
   3. *Fecundity ogive*: Logistic model that needs Gestation period (GP) and maturity parameters slope and intercept.
   4. *mx*: Part of Euler-Lotka equation giving number of offspring produced by each age class. Requires the number of female offspring (n.pups)
   5. tmax: Uses the von B K parameter to get maximum age
   6. Natural mortality: This is done using 5 different methods, with the option to pick any one of them at random that doesn’t really work I don’t think.…
      1. Jensen mat – Needs age a maturity -
      2. Jensen k – Needs the k parameter from the Von B relationship -
      3. Then-hoenig – Needs the maximum longevity (tmax) which seems to rely on k from Von B.
      4. Then-Pauly – Used L\_inf and k to get at natural mortality
      5. Peterson – Uses the Weight for each age, I think give age specific natural mortality
   7. What we return from the parms.calc() function are
      1. age.mat.vec: The age a maturity, one value for each simulation
      2. max.age.vec: Vector of max ages, one value for each simulation
      3. mx.list: The Euler-Lotka fecundity vector, each age class gets a value, and there is one vector for each simulation
      4. r.M.list: Natural mortality vector/value (can be one value or a vector depending on method), one for each simulation
4. Once we have the life history parameters we can now get at the consequences of these life history parameters on theoretical sustainable levels of exploitation for our population, this is informed by life history and removals
5. The first step in this is to understand how varying levels of fishing mortality will impact population, main question I think here is was level of F will result in the population exactly replacing itself (on average and over time). Welcome to F\_crit.r()
   1. First in this function we have one of several minimize() functions we’ll encounter. This one figures out what value of fishing mortality results in the number of spawners produced per female recruit to be equal to one. This value is F\_crit().
      1. The current form of relative fishing mortality on each age class is going to need to be revised, I believe in the current set up only juveniles are being harvested.
      2. Could we want ability to explore other values to minimize against other than 1?
   2. The key outputs from this function are
      1. The vector of exploitation rates by age
      2. The Number of individuals in the population at F.crit (N.crit)
      3. Several other related life history components come out of this, but I don’t believe we use them for anything.
6. The next step is to figure out how rapidly this population can grow (i.e. population growth, not individual growth). This is done using the old Euler-Lotka equation (Lotka\_r.r)
   1. We again run into another cute little minimize() function. This one is taking the survival and fecundity vectors (by age) and figuring out what the growth of the population is. At this stage we’re pretending f =0 and just figuring out what r is without fishing (rmax)
      1. What we need here is age at maturity, maximum age and natural mortality
         1. When using this to figure out r at various levels of fishing we also need exploiation rate at selectivity
      2. In the general function we get the reproductive contribution of each age class (lxmx) and use that to solve for r. The main object returned here is the value of r, which is returned in the ‘par’ slot of the minimizer.
7. OK, now that we have all of this in place we are ready to do the backward simulation on the stock. This, I think, is really what we need for the paper. In the above there has been some stochasticity in the life history parameters, which is the only reason that we can do simulations
   1. The most important piece in here is to have an estimate of the numbers of individuals at the end of the time series. Basically this is the initial condition the model needs.
   2. We start by simulating what the population history would have been in the absence of fishing, using rmax from the Euler-Lotka calcs
   3. We use a log-linear model to get the population growth rate for each simulation over the course of the time series
   4. Next we get the estimated fishing mortality in each year from our simulation using the u.calc.r() function
      1. We have another minimize() function in here, this function finds the value of fishing mortality that leads to the observed removals. As with F crit there are some assumptions in here we’ll need to tidy up.
      2. The time series of exploitation rate is what comes out from this.
8. After this, we then use u.cacl.r() to estimate exploitation rate in the final year of the backwards simulation using various ‘future’ removals scenarios.
9. Next up we do forward projections, done in much the same way as the above. I’m not 100% sure if have any reason to do the forward projections for this project, so we should talk about this.
   1. Start by getting r (lotka.r) and N.crit (F.crit.r) for the various scenarios
   2. There is an option to use historical removals to get the r estimates rather than specifying the removals scenarios Do we want to have this option
   3. There is an option to use autocorrelated deviates of r (using sigma) to project population growth into future. Do we want to have this option
   4. We then use the last years population abundance to start the forward projections, this doesn’t account for potential variability in r (which Heather says is “lazy”, so do we not want to be lazy?).
   5. We project forward for each of the various scenarios using a logistic growth model, where K is current set so high that it doesn’t really do anything, so is basically exponential growth. How to handle K.
10. And we’re basically done this part of the show, all the data are output and ready to be plotted.

**But wait there is more!**

1. There is the plot function (plot\_por\_sim.r), which can plot our output, but also does way more than the function name suggests because it has standard\_SHK.r() embedded in it. Basically, this function will plot our figures, but also, quasi-secretly, do some cool shit to see when/if the stock is being overfished.
2. So the plot function starts by organizing the output from the simulations you’ve run
3. Then it goes and runs standard.SHK() which is a Monte Carlo simulation step that uses a paper by [Brooks and friends from 2010](https://academic.oup.com/icesjms/article/67/1/165/595670?login=true) to develop reference points and enables us to see what the ‘stock’ status for our stock appears to be.
   1. The inputs to standard.SHK() are same as inputs for parms.calc() because they are basically the same function, just this one has some extra bells and whistles on it.
   2. The Monte Carlo simulation takes the life history info you have as input and gets the life history parameters we need, just like parms.calc() does, but it does that within the Monte Carlo simulation, some key new parameters it calculates
      1. Net Reproducitve Rate (fecundity times survivorship)
      2. Generation time
      3. Maximum lifetime reproduction rate
      4. Steepness (Calculation from Brooks 2010 originates in a RAM 1999 paper of course)
      5. Population reproductive rate of some sort (I don’t know where that calculation comes from)
      6. SPRmer Spawners per recruit at maximum excess recruitment, this is the reference point from Brooks et al. 2010
4. Once all that fun stuff is calculated we end up back in the plot function where we make a couple of figures
   1. The first figure (future\_projections.png) simply plots the median Number of individuals over time for our future projections in each of the removals scenarios
   2. The second figure (Abundance\_removals\_ts.png) plots the historic time series with 80% CI’s on top panel, and Removals time series on bottom panel
   3. The 3rd figure relies on the output of the reference point calculations to make a plot to see how frequently the MC simulation suggests our stock is overfished (high\_productivity\_overfished.png). Lots of stuff around this figure and the data going into it that we should discuss
   4. Some of the results are summarized in a few csv files
   5. Then the final plot is produced, this one standardizes the biomass by the Biomass at the SPRmer if we are above 1 we aren’t overfishing, if we go below 1 we are overfishing. Again this is a forward projection figure (b\_bsprmer.projection.png)

**Table of Inputs for functions**

| **Name** | **Type** | **Function used** | **Description** |
| --- | --- | --- | --- |
| N.options | Input | Por.sim | An estimate of the catch/landings/removals from the fishery. |
| n.sims | User | Por.sim | How many simulations do you want to run |
| q.ext | Value | Por.sim | Population extinction threshold, basically a value after which the population is probably screwed |
| Sigma | Parameter | Por.sim | Used if you are assuming autocorrelated values of r for the population. Set to 0 means r is random each year |
| u.long | User | Por.sim | This will select more (but still not all how it is currently coded) of the exploitation data for the analysis |
| Repro.cycle | Input | Por.sim | The number of times an individual reproduces in a year |
| Linf.mn | Input | Parms\_calc  standard\_SHK | The mean estimate of the Von B asymptote. |
| k.mn | Input | Parms\_calc  standard\_SHK | The mean estimate of the von B rate parameter. |
| t0.mn | Input | Parms\_calc  standard\_SHK | The mean estimate of the von B age at size 0. |
| Linf.se | Input | Parms\_calc  standard\_SHK | The standard error of the the Von B asymptote. |
| k.se | Input | Parms\_calc  standard\_SHK | The standard error of the the Von B rate parameter. |
| t0.se | Input | Parms\_calc  standard\_SHK | The standard error of the the Von B age at size 0. |
| mat.int.mn | Input | Parms\_calc  standard\_SHK | The mean of the maturity model intercept. |
| mat.slop.mn | Input | Parms\_calc  standard\_SHK | The mean of the maturity model slope |
| mat.int.se | Input | Parms\_calc  standard\_SHK | The standard error of the maturity model intercept. |
| mat.slope.se | Input | Parms\_calc  standard\_SHK | The standard error of the maturity model slope. |
| W.a | Input | Parms\_calc  standard\_SHK | The Weight-length relationship intercept. |
| W.b | Input | Parms\_calc  standard\_SHK | The Weight-length relationship exponent. |
| n.offspring | Input | Parms\_calc  standard\_SHK  Lotka\_r | Average number of offspring which is transformed to number of female progeny per female per year. |
| gest.period | Input | Parms\_calc  standard\_SHK | The gestation period in years. |
| max.age.lb | Input | Parms\_calc  standard\_SHK | The lowest likely 'maximum age'. |
| repro.cycle | Input | Parms\_calc  standard\_SHK  Plot\_por\_sim | The number of times you reproduce in a year. |
| vb.cor.mat | Parameter | Parms\_calc  standard\_SHK | The correlation matrix for the von B parameters to ensure that we don't get weird parameter combinations. |
| mat.cor.mat | Parameter | Parms\_calc  standard\_SHK | The correlation matrix for the maturity parameters to ensure that we don't get weird parameter combinations. |
| age.mat | Output | F\_crit  U\_calc  Lotka\_r | The age at maturity estimated in parms\_calc using life history information |
| max.age | Output | F\_crit  U\_calc  Lotka\_r | The longevity of the species estimated in parms\_calc using life history information |
| mx | Output | F\_crit  Lotka\_r | The fecundity ogive, so this integrates age at maturity and weight information to sort out how many, estimated in parms\_calc using life history information |
| r.M | Output | F\_crit  U\_calc  Lotka\_r | The natural mortality estimated in parms\_calc using life history information |
| sel | Input | F\_crit  U\_calc  Lotka\_r | The selectivity of the stock. Currently this is set up as a single number and it is tweaked in the code, we'll need to make this more complex. |
| removals | Input | F.crit  U\_calc | Removals from the fishery in a given year. The can originate in the N.options object if doing the backwards simulation, but variable we input for future scenarios. |
| N | Output | U\_calc | Population size in a given year, calculated |
| u | Input | Lotka\_r | Exploitation rate (annual not instantaneous), currently set up to be 1 value. |
| r.cutoff | Parameter | Por\_sim  Lotka\_r | The maximum value we’ll allow for the Lotka\_r() calculations, do we want this for fish? |
| K | Parameter | Por\_sim | Used in the forward projections where we assume a logistic growth model, going to be hard to estimate for fish me thinks? |
| n.mc | Input | standard\_SHK  plot.por.sim | The number of Monte Carlo simulations to run |
| AAFC | Parameter | standard\_SHK | I think Age At First Capture. Seems like this assumes selectivity to be 1 |
| F | Input | standard\_SHK | Fishing mortality from the future removals scenarios, one for each scenario. |
| dat | Input/Output | plot\_por\_sim | Simulation results from por.sim() |
| year | Input | plot\_por\_sim | The initial year of the forward projections, I believe we drop this one from the output results |
| gest | Input | Lotka\_r | The gestation period, for fish well make this 1 (i.e. this will effectively enable an individual to reproduce once a year |
| juv.mult | Input | Lotka\_r | Natural mortality multiplier for Juveniles, will default to 1 |
| repo.lag | Input | Lotka\_r | Lag in years between giving birth and conceiving again. Likely 0 for most fish. |

**Questions on above table**

1. u.long – I think it’d be useful to have this kinda term in there, but needs to be generalized so we can pick out the removal years we want to use.
2. Going to wrap all of the “Input” life history parameters into one big object, might be easiest to put them all into a ‘database’ (i.e. spreadsheet) so we can just call that in, though possible that we could use the Thorson FishLife package to pull right from Fishbase
3. n.offspring – So what do we want here for fish, any way we can get around this given how uncertain it would be I wonder?
4. vb.cor.mat & mat.cor.mat – Can we just stick with what we have for the matrix, or do we need to have a few options here? Seems like there would be some general ones

**Notes in functions**

* (Por\_sim) How should we handle removals, in the example we remove certain years, do we want to have that ability for our generalized model? See code around mean.removals <-
* (Por\_sim) The selectivity is just 1 number and is hard coded to be age 2 on wards we are going to need to generalize this and make a vector. See code around sel <- 1
* (Por\_sim) The carrying capacity…. Do we care since this is just used in the forward projections. If we do care, we’ll need to discuss how to parameterize that, maybe maximum Number ever observed.
* (Por\_sim) N.end, ideas on how we specify this in the absence of an assessment, if we use the last value of the assessment, then our result isn't independent of the assessment, that ok? Could we use a q corrected survey biomass value to kick it off if we didn’t have an assessment model? See code around N.end <- c(sample…
* (Por\_sim) There is a comment saying… 2 is a patch to get n.sims values for r that are <r.cutoff… not sure what that means.
* (Por\_sim) Would we ever want to grab the uncertainty around the B1.vec from the linear model… lm(log(Pop.vec)~years)$coef[2]
* (Por\_sim) If we care about the future simulations… We'll need to make an argument for the function so we can alter the future year removals scenarios, we'll probably want an option to use these, or use the above 'junk' code
* (Por\_sim) Question about using the u.long. We don’t use that right now, but we may want to if we want to use past removals to predict future removals, see code around if(u.long==T). Do we want to keep this u.long at all, or just change it so we can specify in the input what years we want to pull
* (Por\_sim) Question about whether we want to mess around with having autocorrelated r values for the simulation, see code around if(sigma>0)
* (Por\_sim) We use the last 3 years of removals to get estimate for year 1 of the future simulations. We may want to use something other than the removals in the final 3 years, so might make this an option. See code around R.ave
* (Por\_sim) In the forward projections we assume that the population grows and then the removals come after growth (so effectively at the end of the year), would this be useful to generalize so we could have removals then growth? See code around Forward.pop.vec[1]
* (Por\_sim) Do we want to put some uncertainty around r in the forward projections. Se code around xx <- as.vector(r.scenarios[[i]])
* (Several) It looks to me like the fishing mortality age vector is set up to only harvest immature ages and there is no harvest of mature individuals, am I missing something or is this a unique things for Porbeagle?
* (Parms\_calc) Where does this come from, note is uses k from the Von B
* (Parms\_calc) I think the mortality table on line 133 is always just taking the first column so always picks the Jensen.k value
* (F\_crit) Heather can you explain the advantage of assigning these as the dot versions cause I never have wrapped my head around it long enough for it to stick
* (F\_crit) related, should age.mat and max.age be .age.mat and .max.age on line 26
* (F\_crit) Line 35, I think we’ll need to generalize this, I think this is saying Fcrit occurs where each mama produced one recruit, the fishmethods sbpr function does mostly the same thing as this, but you can define what you are minimizing against (e.g. instead of 1 make it a % of MSP)
* (F\_crit) Line 44, This is a vector of f values for each age class, how this line is set up, only immature individuals are given an F.crit, which certainly won’t be our situation
* (standard\_SHK) Line 166 issue in Params Calc about the M.table not really doing what is expected
* (standard\_SHK) Line 177 issue from F\_crit about the F vector only selecting immature individuals.
* (standard\_SHK) Line 181, I don’t think the if-else is needed because in the f.vec you have set F = 0 below AAFC (Age At First Capture)\
* (standard\_SHK) Line 230, I think the alpha.hat estimate is relying on assuming a constant M across age classes, is it ok to have M vary (time or by age) here? There is a SPR method in fishMethods package that can handle variable M, wondering if we should explore that? I probably should read Brooks 2010 - Analytical reference points for age-structured models: application to data-poor fisheries
* (standard\_SHK) Line 235, Where does this equation come from?
* (plot\_por\_sim.r) Line 23 and 36 are not used I don’t think?
* (plot\_por\_sim.r) Lines 100-113 – Why do this section? Also line 105 There is something weird here as this gives just one value, why make a historgram for 1 value? Should the median() go away? Also line 108 I don't see point of this figure or this bit of cod?
* (plot\_por\_sim.r) Line 121 - So do we just assume first year of our data is 'unfished', that seems likely problematic for a bunch of stocks? Maybe it don't matter for our purposes?
* (plot\_por\_sim.r) Line 133: I'm not sure why 0.5 is used, just 50% of MSY p?
* (plot\_por\_sim.r) Line 188 So the geom\_hline has SPRmer.S0/SPRmer.S0... why not just put 1?