

Mathematics Behind The New Mizer

Richard Southwell

August 31, 2018

set__scaling__model()

set__scaling__model() is a procedure in mizer for making a size spectrum model.

Inputs

The scale invariant size spectrum model generated by *set__scaling__model()* depends upon the following inputs:

- The number of species $no_sp = s$. By default this is 11.
- The minimum asymptotic size $min_w_inf = W_\infty^L = \min \{W_{\infty,i} : i \in \{1, \dots, s\}\}$. By default this is 10
- The maximum asymptotic size $max_w_inf = W_\infty^H = \max \{W_{\infty,i} : i \in \{1, \dots, s\}\}$. By default this is 10^3
- The minimum egg size $min_egg = w_e^L = \min \{w_{e,i} : i \in \{1, \dots, s\}\}$. By default this is 10^{-4}
- The minimum maturity weight $min_w_mat = w_*^L = \min \{w_{*i} : i \in \{1, \dots, s\}\}$. By default this is $10^{0.4}$
- The number of weight bins $no_w = K$. By default this is $1 + 100 \log_{10} \left(\frac{W_\infty^H}{w_e^L} \right)$
- The minimum weight of plankton modeled, W_{min}^P min_w_pp is set to $\frac{\min \{w_{e,i} : i \in \{1, \dots, s\}\}}{\beta e^{\delta \sigma}}$ by default.
- The maximum weight of modeled plankton $w_pp_cutoff = w_{cut}$. By default this is W_∞^L . [We ensure that the plankton's carrying capacity $c_p(w)$ has the feature that $w > w_{cut}$ implies $c_p(w) = 0$.]
- The exponent of the maximum consumption rate (also known as the ‘the scaling of intake’), n . By default this is $2/3$.
- The scaling of standard metabolism, q . By default this is $3/4$

- The exponent λ of the background spectrum. By default this is $2 + q - n$
- The growth rate of the primary productivity (the background spectrum), $r_{pp}=r_0$. By default this is 0.1.
- The pre-multiplier of the community spectrum power law κ . By default this is 0.005.
- The assimilation efficiency α . By default this is 0.4
- Multiplicative constant for std. metabolism and activity, k_s . By default this is 4
- Multiplicative constant for max. food intake, h . By default this is 30.
- Preferred predator prey mass ratio β . By default this is 100.
- Width of prey size preference σ . By default this is 1.3.
- Initial feeding level f_0 . By default this is 0.6
- The size beyond which the knife edge gear catches (if that is what is setup for fishing), `knife_edge_size`. The default for this is 100 [I think this is a weight, and not a length, any may not be relevant depending on the gear names].
- The name of the gear, `gear_names`. By default this is "knife_edge_gear", indicating to mizer to setup the fishing selectivity using knife edge gears.
- The factor R_{fac} such that $R_{max.i} = R_i R_{fac}$, for each species i . Here R_{max} is the maximum recruitment allowed and R_i is the steady-state recruitment. [Thus the larger R_{fac} the less the impact of the non-linear stock-recruitment curve.] By default this is ∞ .
- A Boolean variable 'perfect'. By default `perfect=FALSE`. If `perfect=TRUE` then parameters are set so that the community abundance, growth before reproduction and death are perfect power laws.

Procedure

Tidy up inputs and check validity

- If community exponent λ was given as an input then the exponent for volumetric search rate q is rewritten so $q := \lambda - 2 + n$, where n is the exponent for max. food intake.
- If $R_{fac} \leq 1$ then $R_{fac} := 1.01$.
- The number of weight bins K is replaced with the nearest integer to K

- Let $T := \text{round}\left(5 \log_{10}\left(\frac{W_{\text{inf}}^H}{w_e^L}\right) + 1\right)$. If $K < T$ then $K := T$ [this is done so there are at least 5 bins in the interval from w to $10w$ [is this true for any w , or only $w=w_e$ etc. ?]].
- Check that $K \leq 10,000$ and $W_{\infty}^L < W_{\infty}^H$ and $w_e^L < w_*^L$ and $s \geq 2$ and $n, q, r_{\text{pp}}, \text{kappa}, \text{alpha}, h, \text{beta}, \text{sigma}, \text{ks}, \text{f0}, \text{knife_edge_size} > 0$. If any of these conditions are false then give a warning message, and halt.
- If $\text{perfect} = \text{TRUE}$ then let $w_{\text{cut}} := \infty$

Define variables

- Set the exponent of standard metabolism to be $p := n$
- $\lambda := 2 + q - n$
- Define the smallest weight we model fish at to be $W^L := w_e^L$.
- Define the largest weight we model fish at to be $W^H := W_{\infty}^H$
- To round the minimum maturity weight w_*^L up to a grid point we let $\delta = \frac{\log_{10}(W^H) - \log_{10}(W^L)}{K-1}$ and $j = 1 + \left\lceil \frac{\log_{10}(w_*^L) - \log_{10}(W^L)}{\delta} \right\rceil$ and $v = 10^{\log_{10}(W^L) + (j-1)\delta}$ and we round the minimum maturity weight up by letting $w_*^L := v$.
- To round the minimum asymptotic weight w_{∞}^L to the nearest grid point we and $J = 1 + \left\lceil \frac{\log_{10}(W^H) - \log_{10}(W_{\infty}^L)}{\delta(s-1)} \right\rceil$ where s is the number of species. We let $W_{\infty}^L := 10^{\log_{10}(W^H) - J(s-1)\delta}$. [actually is a $\text{round}()$ function used here instead ? maybe we should use the same approach in more places].
- Prepare to setup egg weights: Let $I[e, i] \in \{1, \dots, K\}$ denote the index of the weights that corresponds with the weight such that $w_{I[e, i]} = w_{e, i}$. We set $(I[e, 1], I[e, 2], \dots, I[e, s]) := (1, J+1, 2J+1, \dots, (s-1)J+1)$.
- We set the maximum egg size to be $w_e^H = w_e^L \left(\frac{W^H}{W_{\infty}^L}\right)$.
- Determine logarithmic spacing of egg weights $\Delta = \frac{\log_{10}(w_e^H) - \log_{10}(w_e^L)}{s-1}$.
- Let $(x_{e,1}, x_{e,2}, \dots, x_{e,s}) = (\log_{10}(w_e^L), \log_{10}(w_e^L) + \Delta, \dots, \log_{10}(w_e^L) + (s-1)\Delta)$. The egg weight of background species i is set to $w_{e,i} := 10^{x_{e,i}}$.
- The asymptotic weight of background species i is set to $W_{\infty,i} := w_{e,i} \left(\frac{W_{\infty}^L}{w_e^L}\right)$.
- The maturity weight of background species i is set to $w_{*i} := w_{e,i} \left(\frac{w_*^L}{w_e^L}\right)$.
- The value of the reproduction efficiency is set to $\epsilon = 0.1$, although it is overwritten later.

Setting up the data structure

- A data-frame \mathbb{F} is made holding the species-specific information, together with a declaration that $z_0 = 0$, [where z_0 is the pre-multiplier of mizer's traditional background death term I presume], however we do use a different type of background death term that is described later. The other information included in the data frame are the names of the species, 1, ..., s and w_min , w_inf , w_mat , w_min_idx , h , ks , β , σ , α , ere_pro , $sel_func = "knife_edge"$, $knife_edge_size = knife_edge_size$, $gear = gear_names$.
- A mizer params object \mathbb{P} is made by using \mathbb{F} and $p, n, q, \lambda, f_0, \kappa, W^L, W^H, K, W_{min}^P, w_{cut}, r_0$ as inputs to `MizerParams()`. Here W_{min}^P is the minimum weight of plankton modeled, and r_0 is the replenishment rate of the plankton. The values of γ, w and dw are (re)computed by calling `MizerParams()`, and we use such updated values.

Steady state solution for species one

- Compute coefficient of predation death rate that would be induced by a power law spectrum of background species: $\mu_0 := (1-f_0)\sqrt{2\pi\kappa\gamma}\sigma\beta^{n-1}\exp(\sigma^2(n-1)^2/2)$
- Compute coefficient of energy available for reproduction and growth, at power law state $\hbar := \alpha h f_0 - k_s$. If $\hbar < 0$ then halt.
- We define $pow=B := \frac{\mu_0}{\hbar(1-n)}$, and if $B < 1$ then an error message is displayed about how at steady state the ratio of death rate to growth rate is too small, leading to an accumulation of fish at their largest size.[check the 1-n really goes on the denominator].
- The death rate at steady state is $\mu(w) := \mu_0 w^{n-1}$
- The growth rate of species 1 at steady state is $g(w) := \hbar w^n (1 - \psi_1(w))$, where $\psi_1(w)$ is the fraction of their energy that a weight w member of species 1 diverts to reproduction.
- Let $I[\infty, i]$ denote the index such that $w_{I[\infty, i]} = W_{\infty, i}$, that gives the asymptotic weight of the i th species. Let $\mathbb{I} = \{1, 2, \dots, I[\infty, 1] - 1\}$
- The unscaled steady state abundance $\widehat{N}_1(w)$ of species 1 is such that $\widehat{N}_1(w_1) = 1$, and for each $j \in \{2, \dots, K\}$ we have that $\widehat{N}_1(w_j) = 0$ if $j > I[\infty, 1]$ and otherwise, $(\widehat{N}_1(w_1), \widehat{N}_1(w_2), \widehat{N}_1(w_3), \dots, \widehat{N}_1(w_K))$ is set equal to $(1, H[1], H[2], \dots, H[I[\infty, 1]-1], 0, 0, \dots, 0)$. Here $H[k] == \prod_{u=1}^k \frac{g(w_u)}{g(w_{u+1}) + \mu(w_{u+1})dw_{u+1}}$ [This is to match with the steady state of the upwind difference scheme, but I'd better check I've got the indices correct, etc., and simplify].
- Rescale the abundance of species 1 so it lies on a power law. Let $M = \kappa / \sum_{j=1}^{I[\infty, 1]} \widehat{N}_1(w_j) w_j^{\lambda-1} dw_j$ be the multiplier, and generate the properly

scaled abundance of species 1, which we set to be $N_1^*(w) := \widehat{N}_1(w)(10^{\Delta(1-\lambda)/2} - 10^{-\Delta(1-\lambda)/2})M/(1-\lambda)$. [I am not sure where the $(10^{\Delta(1-\lambda)/2} - 10^{-\Delta(1-\lambda)/2})$ renormalization factor comes from]

Finish computing steady state solution

- Initially set $N_i(w_j) = 0$
- For each species $i \in \{1, \dots, s\}$ let $I[e, i]$ be the egg weight index, so $w_{I[e, i]} = w_{e.i}$ and the list of indices which we shall fill abundance values into is $\mathbb{I} = (I[e, i], I[e, i] + 1, \dots, I[e, i] + \|N_1^*\| - 1)$, where $\|N_1^*\| = I[\infty, 1]$ is the number of size bins that get (potentially) non-zero abundance values loaded into them, during the previous construction of the steady state solution.
- Now we use our scale invariant transformation to construct the steady state abundance of the i th species. This is done by setting $N_i(w_{\mathbb{I}[k]}) = N_1^*(w_k) \left(\frac{w_{e,1}}{w_{e,i}} \right)^\lambda$ for each $k \in \{1, \dots, |\mathbb{I}|\}$, where $|\mathbb{I}|$ is the number of indices in our list.
- We compute the community spectrum $N_c(w) := \sum_{i=1}^s N_i(w)$ and save it as part of the MizerParams object \mathbb{Q} that we will return from this procedure.

Setup Plankton

- Define the ‘plankton vector’ to be $N^P(w) = \kappa w^{-\lambda} - N_c(w)$ to be the required abundance to fill up to the power law, over the interval of weights where fish are modeled.
- Cutoff the plankton vector above w_{cut} so if $w \geq w_{cut}$ then $N^P(w) = 0$. If perfect=FALSE then any negative entries generated in $N^P(w)$ are replaced with zeros.
- Let the abundance of plankton (that will be) at steady state $N_R(w')$ be replaced with $N^P(w')$ for each weight w' in the fish weight interval.
- The carrying capacity $c_p(w)$ is defined from the minimum plankton weight, all the way to the maximum fish size. For the weights w' over the sizes at which fish are modeled we set $c_p(w') = N^P(w') \frac{r(w') + \mu_R(w')}{r(w')}$ where $r(w) = r_0 w^{p-1}$ is the replenishment rate of the plankton, and $\mu_R(w')$ is the death rate on the plankton, in the presence of fish with abundance as described by $N_i(w)$.

Setup background death

- Let $\mu_{P,i}(w)$ be the predation mortality rate.

- For each species i set the background death to be $\mu_{B,i}(w) = \mu_0 w^{n-1} - \mu_{P,i}(w)$, and if perfect=FALSE then replace negative background death values with zeros.

Make egg count match newborn abundance

- Let $R_{p,i} = \frac{\epsilon}{2w_{e,i}} \int_0^\infty N_i(w) E_{r,i}(w) \psi_i(w) dw$ denote the total production of eggs for breeders of species i .
- Let $g_i(w)$ be the growth rate of species i
- Let $\mu_i(w)$ be the total death rate of species i
- For each species i , we replace the value of the reproductive efficiency with the value $\epsilon_i^* = \epsilon_i N_i(w_{e,i})(g_i(w_{e,i}) + dw_{I[e,i]}\mu_i(w_{e,i}))/R_{p,i}$ where ϵ_i is the old value of the reproduction efficiency.
- If $R_{fac} = \infty$ then ϵ_i^* is used as the value of the reproduction efficiency. Otherwise, if R_{fac} is finite then the final value of the reproduction efficiency is $\left(\frac{R_{fac}}{R_{fac}-1}\right) \epsilon_i^*$. This is to compensate for using a stock recruitment relationship.
- Add $N_R(w), N_i(w), \epsilon$ to our MizerParams object \mathbb{P}
- Use these values to find $R_{p,i}^* = \frac{\epsilon}{2w_{e,i}} \int_0^\infty N_i(w) E_{r,i}(w) \psi_i(w) dw$ and reset the stock recruitment relationship so $R_{max,i} = (R_{fac} - 1)R_{p,i}^*$. [am a bit confused about rfac]
- Return \mathbb{P}

retune__abundance()

Inputs

- A MizerParams object \mathbb{P}
- A Boolean vector called 'retune', that determines whether a species can be retuned or not. Let $L \subseteq \{1, \dots, s\}$ denote the set of species i that we want to retune the abundance multipliers of, to make the aggregate abundance fit the power law.
- A cutoff value C . A species i with an abundance $N_i(w_{*i})$ at maturity size w_{*i} will be removed if $N_i(w_{*i}) < CN_C(w)$, where $N_C(w) = \sum_{i=1}^s N_i(w)$. The default value is 10^{-3} .

Procedure

- If no species i are such that $\text{retune}[i]=\text{TRUE}$ then return \mathbb{P} and halt.
- Let \mathbb{I}^L and \mathbb{I}^H be the weight indices such that $w[\mathbb{I}^L] = \min \{w_{*i}\}$ and $w[\mathbb{I}^H] = \max \{W_{\infty,i}\}$ that are the limits between which we try to match the near power law $N_C(w)$ with the aggregate renormalized species abundance curves between the maturity size of the smallest species and the maximum size of the largest species. More precisely, we find the abundance multipliers A_i so that the integral of the square of the relative $(\sum_{i \notin L} A_i N_i(w) + \sum_{i \in L} N_i(w) - N_C(w)) / N_C(w)$ over w , between our limits, is minimized, where L is the set of all retuneable species.
- We just focus on the set $S \subset \{1, \dots, K\}$ of weight indices k such that $N_C(w_k) > 0$. Let $\rho = \sum_{i \in \{1, \dots, K\} - S} N_i(w)$ be the aggregate abundance of species that will not be retuned (using the mizer state saved with \mathbb{P}).
- Construct an $|L| \times |S|$ matrix M such that for $i \in \{1, \dots, |L|\}$ and $j \in \{1, \dots, |S|\}$ let $M_{i,j} = \frac{N_{L[i]}(w_{S[j]})}{N_C(w_{S[j]})}$.
- Let $b(w) = (N_C(w) - \rho) / N_C(w)$.
- Apply singular value decomposition [I should read Numerical Recipes section 15.4.2 to better understand] to M . Let d be a vector containing the singular values of A , sorted in decreasing order. Let $\bar{d} = (1/d_1, 1/d_2, \dots)$ denote the inverses of the singular values. Let u and v be the associated variables from the singular value decomposition. We exclude entries from \bar{d} over 10^8 .
- For the i th retunable species $L[i]$ we have that the abundance multiplier that should be used is $A_{L[i]}^* = x_i$ where $x = yu^T b$ and $y_{ij} = v_{ij} \bar{d}_i$. [I want to understand this better].
- Remove any species i such that $N_i(w_{*i}) < C N_C(w)$, where C is the cutoff.
- Return the resulting MizerParams object, with the newly setup near steady state.

[I am wondering if the code could be improved by adding a function that retunes the abundance multipliers so that the death rates are like those in the scale invariant system. A viable modelling philosophy might be to begin by setting up the real world model, with proper fishing etc., but in the initial model, just initially use background species, setup to look like an average fish in the system. Then we can add the explicit species. We already have a function that retunes the abundance multipliers of the background species to get a power law and ensure (among other things) that the background species experience the same growth rates in the new, heterogeneous system, as they did in the original, purely scale invariant system. I think it might be a good idea to include an extra procedure, that retunes the abundance multipliers of the background species to

try and make it so that background species experience the same death rates as they did in the original system. So basically I am saying, if we get a ‘pure background’ model setup properly first, then we could try and add the explicit species in such a way that the species in the pure background system experience the same growth and death rates as they did in the ‘pure background’ model. Perhaps this idea is misguided, or in some way I have not fathomed yet, we are already doing it. I will be able to say more when I have finished the code review. Also, preserving the death rates of the background species corresponds with an extra constraint that we may not want to have, and there could be a tricky aspect to joint optimizing over the growth and death rates. Anyhow, I’m not sure if this could help us improve the accuracy of our growth rates and SSB values]

[I wonder if it could be helpful to add another procedure $\text{morph}(\mathbb{P}, \mathbb{Q})$ to mizer, that takes a system with parameters \mathbb{P} , at steady state S , and Quasi-continuously deforms the system from \mathbb{P} to system with parameters \mathbb{Q} , along the line from \mathbb{P} to \mathbb{Q} in parameter space, while keeping track of the steady state using $\text{steady}()$. I wonder if we could make the step size involved in the deformation variable to ensure that $\text{steady}()$ has the best change of successfully tracking the steady state.]

addSpecies()

Inputs

- A MizerParams object \mathbb{P} for the system we want to add species to
- A dataframe \mathbb{F} of parameters of the species we want to add.
- A vector S_B where $S_{B,i}$ is the target spawning stock biomass (SSB) of species i . By default these are not specified.
- A value of R_{fac} that determines the strength of the non-linearity in the Beverton-Holt stock-recruitment relationship. The maximal recruitment will be set to R_{fac} times the normal steady-state recruitment. Default is 10.
- The fishing effort e , default value is 0.

Check paramaters are valid

- If $R_{fac} \leq 1$ then change it to 1.01.
- Check no species are being added that are the same as those which are already there.
- Set the reproduction efficiency $\epsilon = 0.1$

- Include line type and line color of species, from \mathbb{F} into the output MizerParams object \mathbb{Q} . Fill out any missing columns of \mathbb{F} with NAs ('not available' markers), and similarly fill out the dataframe of \mathbb{P} .

Make data structure for combined system

- Combine the dataframe \mathbb{F} with the dataframe of the original system \mathbb{P} to get a new dataframe \mathbb{F}' that holds data on all species.
- Use the combined dataframe \mathbb{F}' together with the values $p, n, q, \text{lambda}, f0, \text{kappa}, \text{min_w}, \text{max_w}, \text{no_w}, \text{min_w_pp}, \text{w_pp_cutoff}$ and r_0 [as extracted from rr_pp] and from the original mizer params object \mathbb{P} .
- Use the plankton carrying capacity and plankton abundance $N_R(w)$ from \mathbb{P} . [dont we have to re-adjust the plankton abundance ?]. Add this data to the output \mathbb{Q}
- Let $s+1$ be the index of the new species, and let $s^* = s+1$ be the number of species in the output \mathbb{Q} . [earlier the code suggested that multiple species were being added simultaneously, but actually only one species is being added].
- For each original species $i \in \{1, \dots, s^* - 1\}$, set the abundance $N_i(w)$ of i (associated with the output \mathbb{Q}) initially equal to its abundance in the original system \mathbb{P} .
- Pass the values $N_C(w)$ and $\psi_i(w)$, and $\mu_{b,i}(w)$ and the stock recruitment function (srr), from the original MizerParams object \mathbb{P} into the output \mathbb{Q} . Set the background death for the new species to be the same as the one for the first species in \mathbb{P} . [we are basically assuming the background death terms are the same for the different species, so maybe there should be a check for this. Also, I'm so far a bit unclear how the psi value gets filled in for the new species. Also, I guess srr will be rewritten when we have changed $\text{erepro}, \text{rfac}, \text{rmax}$].

Find near steady conditions for new species

- Set self interaction $\theta_{s**} = 0$, of the new species to zero, so we can determine the growth rates, and death rates induced upon it by the pre-existing species.
- Compute the total death rate for new species $\mu_{s*}(w)$, and the growth rate $g_{s*}(w)$ for the new species, in the system with pre-existing species as previously setup.
- Let $\mathbb{I}[\infty]$ denote the weight index such that $w_{\mathbb{I}[\infty]} = W_{\infty.s*}$, this is the index of the asymptotic weight of the new species.

- Let $\mathbb{I} = (\mathbb{I}[e, s^*], \mathbb{I}[e, s^*] + 1, \dots, \mathbb{I}[\infty] - 1)$ be the weight indices upon which we shall fill in the new species abundace. If $g_{s^*}(w_k) = 0$ for any $k \in \mathbb{I}$ then halt, because algorithm would encounter division by zero. [do we do this when we solve scale invariant system ?]
- Initially fill out the near steady solution $N_{s^*}(w)$ with zeros.
- Let $\mathbb{H} = (I[e, s^*], I[e, s^*] + 1, \dots, I[\infty, s^*])$ denotes the weight indices we shall fill out non-zero values to.
- For each $k \in \{1, \dots, |\mathbb{H}|\}$ let $N_{s^*}(w[\mathbb{H}_k]) := U_k$ where $(U_1, U_2, \dots, U_{|\mathbb{H}|}) = (1, V_1, V_2, \dots, V_{|\mathbb{H}|-1})$ where $V_k = \prod_{u=1}^k \frac{g(w[\mathbb{I}_u])}{g(w[\mathbb{I}_{u+1}]) + \mu(w[\mathbb{I}_{u+1}])dw[\mathbb{I}_{u+1}]}$. This is solving for a steady condition under the Mckendrick von Foerster equation. Check that this candidate steady state does not hold any infinities or non-numerical values.
- If the spawning stock biomass S_B of the new species s^* is not specified then normalise the new species steady state solution so that at its maximum it lies at half the power law, and then calculate its SSB. We choose the maximum of the biomass density in log space because that is always an increasing function at small size. Let \mathbb{I} be the weight index that maximizes $N_{s^*}(w_{\mathbb{I}})w_{\mathbb{I}}^\lambda$ and re-normalize the solution to be $N_{s^*}(w_i) := N_{s^*}(w_i)\kappa w_{\mathbb{I}}^{-\lambda}/(2N_{s^*}(w_{\mathbb{I}}))$, and record the resulting stock spawning biomass $SSB := \sum_k N_{s^*}(w_k)w_k\psi_{s^*}(w_k)dw_k$. Alternatively, if the spawning stock biomass S_B of the new species s^* is $N_{s^*}(w_i) := N_{s^*}(w_i)SSB/SSB^*$.
- The SSB values are recorded as part of a vector \mathbb{A} in the output \mathbb{Q} .
- Reset the self interaction value of $\theta_{s^*s^*} := 1$ [did we check theta started out all ones ?]

Finish setting up new steady state

- Apply `retune()` to recreate the aggregate abundance spectrum of the old params object, allowing all the abundance multipliers to vary in free/background species, as indicated by the vector \mathbb{A} which describes which SSB values are known. [maybe we could use less names for these things].
- Reset the reproduction efficiency $\epsilon_i := \epsilon_i N_i(w_{e,i})(g_i(w_{e,i}) + dw_{I[e,i]} + \mu(w_{e,i}))/R_{p,i}$ as long as unmodified reproductive output $R_{p,i}$ is non-zero, in which case set $\epsilon_i := 0.1$.
- Rewrite the reproduction efficiency to be $\epsilon_i := \epsilon_i/(1 - 1/R_{fac})$ [again I have to better understand this rfac stuff].
- $R_{max,i} = (R_{fac}-1)R_{p,i}^*$ where $R_{p,i}^*$ is the recalculated reproductive output.
- Gather data and output it as MizerParams object \mathbb{Q} .

Project.R

The central PDE evolver behind mizer.

Project()

Inputs

- A mizer params object \mathbb{P} .
- A fishing effort level e , [In the most general form, effort is specified for each gear, and each time step of the simulation. Each gear may target a subset of species. But I guess no species can be targetted by more than one gear]. Default is 0.
- The run time for the simulation T^H . Default is 100 (years)
- The time step size dt used in the discretised PDE solver. Default is 0.1 [although values significantly lower could be used].
- The amount of time T^S between saving PDE.
- The initial fish abundance $N_i^0(w)$, that acts as part of the initial condition. The default is the initial fish abundance associated with the mizer params object \mathbb{P} .
- The initial plankton abundance $N_R^0(w)$, that acts as part of the initial condition. The default is the initial plankton abundance associated with the mizer params object \mathbb{P} .
- An option to include a progress bar when this procedure is used within the shiny app.
-

Initialization

- Check that \mathbb{P} is a valid mizer params object.
- If effort is a vector of length greater than one then check it length and entry names corresponding to the set of gears. Check effort names match gear names. And produce more general time by gear array of fishing efforts [what are ‘recycling rules’, and why is the transpose of effort used ?].