Applying the Management Strategy Evaluation tool {openMSE} to the Antarctic Krill fishery case

Bruno Caneco

5/19/23

Table of contents

# Preface

*[Short intro on krill and its current management]*

The Antarctic krill fishery in FAO Area 48 is currently managed based on precautionary catch limits set by the Commission for the Conservation of Antarctic Marine Living Resources (CCAMLR) and so on…

[Describe CCAMLR’s 3 management principles: (i) stable recruitment; (ii) preservation of ecological system, in particular in relation to predators, and (iii) the effects of fishing must be reversible over a fixed period. Relate these principles with the decision rules (i.e. depletion criteria/rule relates to (i); escapement criter)]

*[Role of MSE in fisheries]*

*[Main objectives]*

This analysis aspires to show how the tool openMSE can be applied to the Krill fishery under a data-limited framework.

*[Main outline]*

The analysis is structured in two parts:

* Part 1: Approximate openMSE to the current management approach
* Part 2: Apply an example MSE to Krill

*[R session info]*

# 1. Recruitment Variability and Natural Mortality

Generating random draws of Recruitment Variance and Natural Mortality under different Proportional Recruitment scenarios

## 1.1 Introduction

Yearly recruitment (), the number of young individuals that enter the population every year, and annual natural mortality (), expressing the fraction of the population dying each year from any non-fishing causes (e.g. disease, competition, predation, etc.) are key parameters in population dynamics models.

Due to challenges in obtaining reliable estimates of krill biomass (and hence numbers) at recruitment, standard krill recruitment modelling is instead based on the proportion of recruits in the population. This method, commonly known as the Proportional Recruitment (PR) model, was firstly proposed by de la Mare (1994) and later expanded by Pavez et al. (2021).

Using the mean and variance of the proportion of recruits estimated from independent surveys, the PR model derives the recruitment natural variability (expressed as a coefficient of variance, ) and natural mortality of the population. Estimated values of and can subsequently be used to generate series of that follow the underlying distribution of proportion of recruits (or, specifically, the odds of recruits).

Additionally, as PR estimates are obtained from a finite number of surveys, the PR framework also provides alternative methods to convey the uncertainty in parameters and . These methods involve generating random draws of and via resampling-based techniques such as parametric bootstrap or inverse probability transform (Pavez et al., 2021).

In this section we generate random draws of and under different scenarios of proportional recruitment, which will be used in population model projections at later stages of the analysis. The PR models and scenarios used here are very similar to those applied in Maschette et al. (2021), with the exception that the proportion of recruits is assumed to be log-Normally distributed (as opposed to inverse-Beta).

Many of the key functions used in this analysis are available in a CCAMLR’s [code repository](https://github.com/ccamlr/Grym_Base_Case/tree/Simulations), which provides a base-case implementation of {GRYM} for the stock assessment of the Antarctic Krill.

## 1.2 Set-up the Proportional Recruitment Scenarios

The PR model requires the following input parameters:

1. The within-year natural mortality structure
2. The quantile function that defines the distribution of recruits
3. The mean and variance estimates of the proportion of recruits obtained from time-series of independent surveys, and the number of surveys conducted
4. The reference age-class to determine the proportion of recruits in model simulations
5. Approximate estimates of and
6. The number of parameter sets to generate

For the purpose of this analysis, PR scenarios are defined by a set of different recruitment proportion estimates (item 3, above). Within-year natural mortality pattern and recruits distribution (items 1 and 2, respectively) remain fixed across all scenarios.

We begin by reading-in estimates of recruitment proportion under each scenario, before specifying the remaining fixed parameters.

### 1.2.1 Loading estimates of recruitment proportion

pr\_scenarios <- read\_xlsx(  
 path = "../part1\_shared\_files/inputs/grym\_parameter\_scenarios.xlsx",   
 sheet = "rec\_proportion\_scenarios"  
) |>  
 select(-ref) |>  
 mutate(  
 pr\_scen\_id = factor(pr\_scen\_id, levels = unique(pr\_scen\_id)),  
 n\_surveys = as.integer(n\_surveys)  
 )

Scenario-specific estimates of the fraction of recruits in the population (**?@tbl-pr-scens**) are a subset of those provided in Maschette et al. (2021). Estimates were calculated from different choices of survey time-series, proportion calculation method, length cut-off thresholds, sample estimates weighting, etc (check Maschette et al., 2021 for further details). Proportion calculation methods are detailed in Maschette and Wotherspoon (2021, Appendix 1).

# table with recruitment proportion under each scenario  
pr\_scenarios |>  
 select(-c(independent\_unit, time\_period)) |>  
 relocate(pr\_scen\_id, Maschete\_description, rprop\_calc\_method, data\_years, further\_details, mn\_Rprop:sd\_Rprop) |>  
 flextable() |>  
 colformat\_double() |>  
 set\_table\_properties(width = 1) |>  
 #set\_table\_properties(width = 1, layout = "autofit") |>  
 width(j = c(1, 3, 4), width = 1.2) |>  
 width(j = c(2, 5), width = 1.5) |>  
 set\_header\_labels(  
 pr\_scen\_id = "PR Scenario ID",  
 data\_years = "Years of data",  
 Maschete\_description = "Label in Maschette et al (2011)",  
 mn\_Rprop = "Mean of recruitment proportion",  
 sd\_Rprop = "SD of recruitment proportion",  
 n\_surveys = "Nr of Surveys",  
 further\_details = "Further details",  
 rprop\_calc\_method = "Calculation method"  
 )

**Table** **:** Survey estimates of recruitment proportion under each scenario

| **PR Scenario ID** | **Label in Maschette et al (2011)** | **Calculation method** | **Years of data** | **Further details** | **Mean of recruitment proportion** | **SD of recruitment proportion** | **Nr of Surveys** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PR-emm21 | (1) Initial values | Mixture analysis | 1977-1993 | Initial values discussed by WG-EMM-2021 | 0.5570 | 0.1260 | 17 |
| PR-amlr | (2) US-AMLR summer | Length threshold (cut-off @ 36mm) | 1992-2011 | Catch weighted PR by survey, strata scaled | 0.4079 | 0.3118 | 20 |
| PR-atlantida | (4) Atlántida survey | Length threshold (cut-off @ 36mm) | 2020 | Catch weighted PR by strata, strata scaled | 0.4281 | 0.1112 | 3 |
| PR-amlr-haul | (6) US-AMLR summer haul-by-haul | Length threshold (cut-off @ 40mm) | 1991-2011 | Catch weighted PR by haul | 0.3630 | 0.3700 | 21 |

### 1.2.2 Specify remaining parameters

We now specify the choices for the other model parameters, which remain unchanged across all the considered PR scenarios.

Note: To ensure consistency in model inputs across multiple stages of the current analysis, GRYM setup parameters have been stored in an external file.

# Load fixed parameters   
pr\_fixed\_pars <- read\_xlsx(  
 path = "../part1\_shared\_files/inputs/grym\_parameter\_scenarios.xlsx",   
 sheet = "fixed\_parameters"  
) |>  
 filter(category %in% c("PR model", "Mortality", "Population model structure", "Simulation options")) |>  
 select(parameter\_tag, value) |>  
 pivot\_wider(names\_from = parameter\_tag) |>  
 select(-c(Fmax, n.years))  
  
pr\_fixed\_pars

# A tibble: 1 × 8  
 pr\_ref\_age pr\_dist M\_within\_year first\_age last\_age age\_plus nsteps n\_iter  
 <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>   
1 2 Log-Normal constant 1 7 No 365 10001

#### Within-year natural mortality pattern

Similarly to Maschette et al. (2021), assuming the impact of natural mortality is constant throughout the year.

# within-year mortality option  
pr\_fixed\_pars$M\_within\_year

[1] "constant"

Calculate the associated final row of unscaled integrated natural mortality, which is used in the PR model to compute survivals.

# modelled age-classes   
Ages <- as.integer(pr\_fixed\_pars$first\_age:pr\_fixed\_pars$last\_age)  
n\_steps <- as.integer(pr\_fixed\_pars$nsteps)  
   
# Within-year natural mortality pattern  
if(pr\_fixed\_pars$M\_within\_year == "constant"){  
 # Assuming the impact of natural mortality is constant throughout the year  
 ms <- matrix(1, n\_steps+1, length(Ages))   
}else{  
 stop("Option for within-year pattern in M not identified ('constant' is the only currently available option)")  
}  
  
# calculate the proportion of an individual time step in the year  
h <- 1/n\_steps  
  
# calculate the unscaled integrate natural mortality of each age (columns)  
# through the year (rows)  
Ms <- ctrapz(ms, h)   
  
# final row of the unscaled integrated natural mortality  
Msf <- final(Ms)  
Msf

[1] 1 1 1 1 1 1 1

#### PR quantile function

Recruitment is assumed to follow a log-Normal distribution. This is a deviation from the approach taken in Maschette et al. (2021), where recruitment was assumed to be inverse-Beta distributed. The rationale behind this decision was to align simulations under {GRYM} with those in the {openMSE} framework, which assumes that recruitment process error (i.e. deviations from expected number of recruits) is log-normally distributed.

Functions qLogNormal() and recLogNormal() are provided [here](https://github.com/ccamlr/Grym_Base_Case/blob/Simulations/3_Code/Source/prfit.R).

pr\_fixed\_pars$pr\_dist

[1] "Log-Normal"

if(pr\_fixed\_pars$pr\_dist == "Log-Normal"){  
 # quantile function of the target distribution for odds of recruitment proportion  
 qdist <- qLogNormal  
 # function to generate nr of recruits under the target distribution  
 recdist <- recLogNormal  
}else if(pr\_fixed\_pars$pr\_dist == "Inverse-Beta"){  
 qdist <- qInverseBeta  
 recdist <- recInverseBeta  
} else if(pr\_fixed\_pars$pr\_dist == "Gamma"){  
 qdist <- qGamma  
 recdist <- recGamma  
} else{  
 stop("Chosen value for distribution of odds of recruits is not valid.")  
}

#### Reference age-class, number of iterations and starting values

As in Krill’s assessment base-case and Maschette et al. (2021), setting the 2nd age-class as the reference class (i.e. recruitment proportion is the fraction that the second age-class is of the population of age-class 2 and older).

# reference age-class  
ref\_age <- as.integer(pr\_fixed\_pars$pr\_ref\_age)  
  
# number of iterations  
niter <- as.integer(pr\_fixed\_pars$n\_iter)  
  
# Starting values, i.e. best guess estimates of M and recruitment CV  
M0 <- 0.6  
CV0 <- 1

## 1.3 Generate parameter draws

In this section we use the PR model to generate random draws of variability in recruitment, expressed as coefficient of variance (), and annual natural mortality () for the scenarios under consideration.

This is achieved using the *inverse probability transform* approach described in Pavez et al. (2021) and implemented in function prfit(), available from krill’s base-case [code repository](https://github.com/ccamlr/Grym_Base_Case/blob/Simulations/3_Code/Source/prfit.R).

At each iteration, the PR model is fitted by determining the properties of the underlying distribution of odds of recruits (), which depend on both and , for a given set of random quantiles of the distribution of . Values of parameters and are adjusted to provide the best fit between model predictions and observed data in terms of mean and variance of recruitment proportion in surveys. The fitting process is repeated niter times with newly generated random quantiles, each time returning a single set of estimates (i.e. draws) of , and associated mean () and variance () of odds of recruit.

### 1.3.1 Prepare model and data for parallel computations

Parallelization is conducted at the iteration level, with each parallel node fitting one PR model at a time. We first define the function gen\_rec\_draws() to apply the PR model fitting function prfit() for the specified number of iterations in parallel, for a given set of PR parameter values.

# Wrapper function to run PR models in parallel via `{furrr}`, using  
# `{progressr}` for progress feedback  
gen\_rec\_draws <- function(niter, mn\_Rprop, sd\_Rprop, n\_surveys, M0, CV0, ref\_age, qdist, Msf){  
  
 p <- progressor(steps = niter)  
  
 future\_map\_dfr(  
 1:niter,   
 function(x){  
 out <- prFit(qdist = qdist, Msf = Msf, mnR = mn\_Rprop, vrR = sd\_Rprop^2,   
 n = n\_surveys, M0 = M0, CV0 = CV0, r = ref\_age)  
 p()  
 out  
 },   
 .options = furrr\_options(seed = TRUE)  
 )  
}

After setting up a reference data frame comprising the parameter values under each scenario, we are ready to run the PR models.

# data.frame with PR inputs under each scenario  
scens\_sim\_pars <- pr\_scenarios |>  
 select(pr\_scen\_id, mn\_Rprop, sd\_Rprop, n\_surveys) |>  
 mutate(M0, CV0, ref\_age, qdist = list(qdist), Msf = list(Msf))

### 1.3.2 Fit PR models for each scenario

The model fitting process is performed sequentially over the considered scenarios, within which multiple fits are performed in parallel via the gen\_rec\_draws() function.

# Note: Model fitting runtime is substantial for some scenarios, so this code chunk  
# must only be executed interactively, i.e. it shouldn't be evaluated during  
# document rendering  
  
if(file.exists("../part1\_shared\_files/outputs/cvR\_M\_draws\_scen.rds")){  
 cat("\n")  
 useropt <- readline(prompt = "File with parameter draws already exists. Want a re-run? ([Y]es or No): ")   
 run\_fitting <- ifelse(str\_starts(useropt, "Y|y"), TRUE, FALSE)  
}else{  
 run\_fitting <- FALSE  
}  
  
if(run\_fitting){  
 # set number of cores to use  
 plan(multisession, workers = availableCores()-2)  
 handlers("progress")  
   
 tic()  
 cvR\_M\_draws\_scen <- scens\_sim\_pars |>  
 rowwise() |>  
 pmap(function(pr\_scen\_id, mn\_Rprop, sd\_Rprop, n\_surveys, M0, CV0, ref\_age, qdist, Msf){  
   
 cli::cli\_alert(glue::glue("Running {pr\_scen\_id}"))  
   
   
 with\_progress(  
 out <- gen\_rec\_draws(niter = niter, mn\_Rprop = mn\_Rprop, sd\_Rprop = sd\_Rprop,   
 n\_surveys = n\_surveys, M0 = M0, CV0 = CV0, ref\_age = ref\_age,   
 qdist = qdist, Msf = Msf)  
 )  
 cli::cli\_alert\_success(glue::glue("Finished {pr\_scen\_id}"))  
 out  
 })  
 toc()  
   
 plan(sequential)  
   
 names(cvR\_M\_draws\_scen) <- scens\_sim\_pars$pr\_scen\_id  
   
 # save draws externally  
 write\_rds(x = cvR\_M\_draws\_scen, "../part1\_shared\_files/outputs/cvR\_M\_draws\_scen.rds")   
}

## 1.4 Outputs checking

Having a look at the outputs for a couple of scenarios, where listed data.frames comprise random draws of , , and for the corresponding PR scenario.

cvR\_M\_draws\_scen$`PR-emm21` |> round(3) |> glimpse()

Rows: 10,001  
Columns: 4  
$ M <dbl> 0.959, 0.909, 0.968, 0.873, 0.757, 0.833, 0.832, 0.846, 0.827, 0.…  
$ CV\_R <dbl> 0.506, 0.670, 0.514, 0.430, 0.754, 0.439, 0.374, 0.411, 0.452, 0.…  
$ mnQ <dbl> 1.615, 1.488, 1.638, 1.402, 1.143, 1.310, 1.306, 1.338, 1.294, 1.…  
$ vrQ <dbl> 0.667, 0.993, 0.707, 0.364, 0.743, 0.330, 0.239, 0.302, 0.343, 0.…

cvR\_M\_draws\_scen$`PR-atlantida` |> round(3) |> glimpse()

Rows: 10,001  
Columns: 4  
$ M <dbl> 0.497, 0.519, 0.660, 0.778, 0.287, 0.609, 0.371, 0.703, 0.652, 0.…  
$ CV\_R <dbl> 0.503, 0.242, 0.469, 0.498, 0.596, 0.483, 0.489, 0.396, 0.446, 0.…  
$ mnQ <dbl> 0.679, 0.712, 0.953, 1.189, 0.405, 0.861, 0.503, 1.036, 0.939, 0.…  
$ vrQ <dbl> 0.116, 0.030, 0.199, 0.350, 0.058, 0.173, 0.060, 0.168, 0.175, 0.…

Next we take a look at the generated draws.

cvR\_M\_draws\_scen\_df |>  
 ggplot(aes(x = M, y = CV\_R)) +  
 geom\_point(size = 0.8, alpha = 0.25) +  
 facet\_wrap(~pr\_scen\_id)

|  |
| --- |
| Figure 1.1: Generated draws of and under each PR scenario |

And complement the with the corresponding summary statistics.

cvR\_M\_draws\_scen\_df |>  
 group\_by(pr\_scen\_id) |>  
 summarise(  
 `Mean of M` = mean(M),   
 `SD of M` = sd(M),  
 #`Median of M` = median(M),  
 # `2.5% of M` = quantile(M, probs = 0.025),  
 # `97.5% of M` = quantile(M, probs = 0.975),   
 `mean of CV\_R` = mean(CV\_R),   
 `sd of CV\_R` = sd(CV\_R))

# A tibble: 4 × 5  
 pr\_scen\_id `Mean of M` `SD of M` `mean of CV\_R` `sd of CV\_R`  
 <fct> <dbl> <dbl> <dbl> <dbl>  
1 PR-emm21 0.853 0.0831 0.510 0.114  
2 PR-amlr 0.771 0.232 4.40 1.93   
3 PR-atlantida 0.612 0.251 0.777 0.940  
4 PR-amlr-haul 0.683 0.252 7.31 1.73

In general, scenarios based on surveys observing higher mean recruitment proportions generate draws (i.e. estimates) of that are, on average, higher. This is expected since higher natural mortality rates lead to a rapid decline of numbers in older age-classes, making the proportion of recruits in the population more prominent. The opposite occurs when natural mortality is low.

Regarding recruitment variability, the magnitude of drawn estimates is, unsurprisingly, correlated with to the variance in observed recruitment proportion under each scenario (see **?@tbl-pr-scens** for reference). Evidently, highly variable yearly recruitment will cause the fraction of recruits in the population to fluctuate strongly from year to year.

The level of uncertainty about parameters and for the considered scenarios is reflected by the spread of the generated draws, which is conveyed by the plots in [Figure 1.1](#fig-M-RCV-draws) and, relatedly, the standard deviation of draws presented in the table above. Results suggest that uncertainty is affected by the variability in recruitment and the number of surveys from which proportion of recruits are estimated (**?@tbl-pr-scens** for reference).

Some further observations:

* draws generated under the “PR-emm21” scenario are notably more condensed compared to the remaining scenarios. This indicates that “PR-emm21” has the lowest uncertainty about both and parameters.
* Despite the relatively low values of drawn under “PR-atlantida”, uncertainty about and for this scenario is larger (i.e. SD is higher) than that obtained under “PR-emm21”. This is likely related to the fact that recruitment proportions were calculated from only 3 surveys (as opposed to 17 surveys under “PR-emm21”).
* Differences in the distribution of draws of and between scenarios “PR-amlr” and “PR-amlr-haul” illustrate the impact of length cutoff thresholds and sample weighting choices applied during the estimation of the recruitment proportion in surveys.
* Draws of for scenarios “PR-amlr” and “PR-amlr-haul” show that yearly recruitments series simulated under these scenarios will have sizable levels of variability (i.e. over 400% and 700%, respectively).

### 1.4.1 Reverse-check

Here we simulate survey estimates of mean and variance of recruitment proportion from the PR model, for each of the considered scenarios. Recruitment series are generated from each draw of and , which are then used to simulate the proportion of recruits that would be seen in surveys. Thus, each draw generates a series of “observed” proportion of recruits from which estimates are calculated.

The core simulating function prSim() is provided in krill’s base case [code repository](https://github.com/ccamlr/Grym_Base_Case/blob/Simulations/3_Code/Source/prfit.R). [Figure 1.2](#fig-hdr-sim-mn-vr-prop-rec) display the simulation results in terms of highest density region (HDR) plots.

scens\_sim\_pars |>  
 mutate(pr\_scen\_id = factor(pr\_scen\_id, levels = unique(pr\_scen\_id))) |>  
 group\_split(pr\_scen\_id) |>  
 map2(.y = cvR\_M\_draws\_scen, function(scn\_pars, scn\_draws){  
   
 # Simulate observation process for each draw of CV\_R and M, i.e. generate  
 # `n` random age structures for each draw of CV\_R and M and estimate mean  
 # and variance of the proportion of recruits that would be observed in surveys  
 pr\_mn\_vr <- map2\_df(  
 scn\_draws$M,   
 scn\_draws$CV\_R,   
 ~prSim(  
 qdist = scn\_pars$qdist, Msf = scn\_pars$Msf[[1]], n = scn\_pars$n\_surveys,   
 M = .x, CV = .y, r = scn\_pars$ref\_age  
 )  
 )|> rename(sim\_mnQ = mnR, sim\_vrQ = vrR)  
   
 title <- scn\_pars$pr\_scen\_id  
   
 # Compare simulated values of mean and variance of proportional recruitment  
 # with those actually obtained from surveys  
 ggplot(pr\_mn\_vr, aes(x = sim\_mnQ, y = sim\_vrQ)) +  
 geom\_hdr() +  
 geom\_point(aes(x = mn\_Rprop, y = sd\_Rprop^2), colour = "red", data = scn\_pars) +  
 labs(title = title) +  
 theme(axis.title = element\_blank(), plot.title = element\_text(size = 10))  
  
 }) |>  
 wrap\_plots(guides = 'collect') |>  
 patchworkGrob() |>  
 gridExtra::grid.arrange(  
 left = "Variance of proportion of recruits",   
 bottom = "Mean of proportion of recruits"  
 )

|  |
| --- |
| Figure 1.2: HDR of simulated means and variances of proportion of recruits based on generated draws of and under each scenario. Red dots are original observed estimates used for parameter generation. |

Results suggest that simulated surveys based on generated parameter draws produce estimates of recruitment proportion that are consistent with those obtained from observed surveys.

### 1.4.2 Exemplify generation of recruitment series

For completeness, we generate a recruitment series from a single draw of PR estimates. Simulated numbers of recruits are log-normally distributed with mean of 1 and coefficient of variance given by the used draw of .

r\_series <- cvR\_M\_draws\_scen |>  
 map(slice, 5) |>  
 bind\_rows(.id = "pr\_scen\_id") |>  
 mutate(pr\_scen\_id = factor(pr\_scen\_id, levels = unique(pr\_scen\_id))) |>  
 rowwise() |>  
 mutate(  
 rec\_series = list(recdist(n = 50000, mn = mnQ, vr = vrQ))  
 ) |>  
 unnest(rec\_series)  
  
r\_series |>  
 ggplot() +  
 geom\_histogram(aes(x = rec\_series), col = "black", fill = "gray88") +  
 labs(x = "Number of recruits") +  
 facet\_wrap(~pr\_scen\_id, scales = "free")

|  |
| --- |
| Figure 1.3: Histogram of recruitment series generated from one draw of parameters, for each scenario |

Comparing drawn value of with summary statistics of the generated recruitment series.

r\_series |>  
 group\_by(pr\_scen\_id, CV\_R) |>  
 summarise(  
 mn\_R\_series = mean(rec\_series),   
 sd\_R\_series = sd(rec\_series),  
 cv\_R\_series = sd\_R\_series/mn\_R\_series,  
 .groups = "drop"  
 )

# A tibble: 4 × 5  
 pr\_scen\_id CV\_R mn\_R\_series sd\_R\_series cv\_R\_series  
 <fct> <dbl> <dbl> <dbl> <dbl>  
1 PR-emm21 0.754 0.999 0.745 0.745  
2 PR-amlr 2.80 1.00 3.08 3.07   
3 PR-atlantida 0.596 1.00 0.596 0.595  
4 PR-amlr-haul 9.94 0.991 7.02 7.08

## 1.5 Wraping up

We now have random draws of recruitment parameters and natural mortality, which convey the uncertainty about those parameters. Generated draws will be used as inputs for the GRYM ([Chapter 2](#sec-grym-sims)) and openMSE ([Chapter 3](#sec-openmse-sims)) tools to run population projections and estimate harvest rate levels that satisfy the sustainability rules currently in place for the management of the krill stock.

# 2. GYM/Grym Analysis

Estimating precautionary harvest rates using GYM/Grym under alternative input values

## 2.1 Introduction

The current management of the Antarctic Krill fishery is based on establishing precautionary catch limits that comply with sustainability principles set out by the Commission for the Conservation of Antarctic Marine Living Resources (CCAMLR, Constable et al., 2000). In practice, these catch limits are set on the basis of a constant long-term precautionary annual yield (), which is determined as the proportion () of the pre-exploitation biomass of the population () that can be harvested annually while ensuring that CCAMRL objectives are achieved:

The estimation of and , which can be though of as the precautionary harvest rate, are major tasks in the assessment of Krill fisheries.

Estimation of relies on a frequentist probability approach using Monte-Carlo simulations. In this approach, a stochastic population dynamics model is used to project the stock forwards in time under a range of potential values, each representing a different fraction of taken as a constant catch in each year of the projection.

For each projection, the population’s pre-exploitation state and key stock parameters (e.g. yearly recruitment, mortality, growth) are drawn at random from suitable statistical distributions that account for the natural variability in the population as well as the uncertainty about those key parameters. Projections are repeated thousands of times for new randomly drawn parameter values to evaluate the probability distribution of population status at each year of the projection period under different levels of yield (and thus, fishing mortality). The projection period covers a minimum of 20 years to ensure that long-term trends in the population can be adequately assessed.

Finally, is determined using the following three-step decision rule (Constable et al., 2000):

1. **Depletion rule**: choose a harvest rate, , so that the probability of the spawning stock biomass () dropping below 20% of its pre-exploitation level () at any given year of a 20-year harvesting period is no more than 10%[[1]](#footnote-50).
2. **Escapement rule**: choose a harvest rate, , so that the median escapement at the end of a 20 year period is 75% of the pre-exploitation median level - i.e. the at which the median of the distribution of after 20 years of harvesting is no less than 75% of the median[[2]](#footnote-52) of .
3. Select the lower of and values (which returns an harvest rate that is consistent with both previous rules).

The relative magnitudes of or depend largely on the level of recruitment variability and the degree of uncertainty associated with the estimate of used in the model (Constable et al., 2000).

### 2.1.1 The GYM/Grym framework

The current precautionary catch limit for Antarctic Krill was determined in 2010 (CCAMLR, 2022) based on a estimated from the Generalized Yield Model (GYM) developed by Constable and de la Mare (1996). GYM is an age-structured, non-spatially explicit, single-stock simulation model that accounts for natural variability and uncertainty on population estimates. GYM is not a statistically fitted stock assessment model - i.e. it does not estimate stock parameters from input data. Therefore, all input parameters must be estimated externally and provided directly to the model.

GYM, originally written in Fortran, was later updated and converted into the R package {Grym} (Maschette et al., 2020), which was used to develop CCAMLR’s [base case](https://github.com/ccamlr/Grym_Base_Case/tree/Simulations) implementation of Krill assessment for management advice purposes. Maschette and Wotherspoon (2021) describe the main input parameters used by GYM/Grym for modelling the Krill fishery.

Most relevant features of the GYM/Grym base case configuration for the Krill stock include:

* Stock is structured into 7 age-classes, and assumes that individuals recruit to the modelled stock at age 1. Therefore, the model covers the age ranges 1-2 up to 7-8 years old.
* The final age-class is not an age-plus group, i.e. it is assumed that no individual lives more than 8 years.
* Stock dynamics are modelled at daily time steps, i.e. the model evaluates the status of the stock at 365 time-points whithin each year of the projection period.
* Recruitment is not dictated by a conventional stock-recruitment relationship, such as the Beverton-Holt or the Ricker functions. Instead, yearly numbers of recruits are simulated as random deviates from a mean recruitment that is constant over time, regardless of the spawning stock size. However, the model includes a depletion factor that reduces the simulated recruitment in a given year when the stock falls below the depletion critical point (i.e. ) in the preceding year. The magnitude of this reduction is proportional to the percentage drop from the critical depletion point.
* The main sources of stochasticity between simulations are:
  + Yearly Recruitment: series are randomly generated from a statistical distribution (Inverse-Beta in the base case) with a fixed mean of 1 and variance estimated from the Proportional Recruitment (PR) model. Recruitment variance, accounting for natural variability in the recruitment process, remains fixed over all years of the projection period, in a given simulation.
  + Recruitment variance: to account for uncertainty in the estimate of recruitment variance, each simulation uses a new random draw of PR estimates, which are generated in the PR analysis ([Chapter 1](#sec-generate-RCV-M)).
  + Natural Mortality: for each simulation, a single random value of annual scaling of natural mortality () is used to simulate the degree of stock decay due to non-fishing causes over the entire projection period (i.e. assumed age-independent and constant over time). Natural mortality is also estimated in the PR model, which provides random draws of accounting for the uncertainty in the estimation process ([Chapter 1](#sec-generate-RCV-M)).
  + Maturity and gear selectivity ogives: the midpoints of the ogive ramps describing, respectively, the length-class at which 50% of its individuals are mature or available to the fishery, are randomly draw in each simulation from a uniform distribution bounded by a selected minimum and maximum lengths. This accounts for the lack of knowledge about the exact length at which 50% of Krill are either mature or selectable by the fishery. Sampled midpoint values are the same in all years of a given projection.
  + Pre-exploitation Biomass Survey Estimate: within each simulation, the annual catch limit under each of the considered values is simulated as , where is a survey estimate of pre-exploitation biomass. To simulate this estimation process, values are generated by applying log-normal errors to the simulated values, with variance reflecting the sampling error in surveys. The input value of parameter is calculated externally from survey data (e.g. Kinzey, 2021).
* In each simulation, sampled values of stochastic parameters are held fixed across projections covering the range of considered values. This approach ensures that observed changes in population forecasts within a given simulation are solely attributable to different fishing pressures and not confounded by randomness underlying the simulation process.
* Grym provides the ability the decompose natural mortality and fishing mortality into intra-year components to account for within-year patterns in mortality incidence. However, under the base-case implementation, natural and fishing mortality are assumed to impact all individuals equally through the year.
* Weight-at-length and length-at-age relationships are assumed to be fully deterministic, meaning that growth is considered to be unaffected by natural variability, while power law parameters and von Bertalanffy growth parameters are treated as known without error.

### 2.1.2 Analysis goal

Here we run the GYM/Grym base-case implementation for 8 alternative scenarios of input parameter values, resulting from combining four different PR estimation scenarios with two options for the maturity-at-length ogive curve. This is similar to the ensemble analysis previously conducted by Maschette et al. (2021), with the exception that here we assume recruitment to follow a Log-Normal process.

Random draws of , recruitment variability and related PR estimates for each of the four PR scenarios were pre-generated in [Chapter 1](#sec-generate-RCV-M).

## 2.2 Estimating under alternative Grym parameter values

### 2.2.1 Specify input scenarios

We begin with setting up the combination of Grym parameter values for each scenario. We consider 3 groupings of input parameters:

1. Recruitment parameters estimated from four different PR scenarios ([Chapter 1](#sec-generate-RCV-M))
2. Maturity-at-length parameters from two 2 alternative maturity ogive curves
3. All remaining input parameters, which take unique values across scenarios.

# Proportional Recruitment scenarios  
prop\_rec\_scenarios <- read\_xlsx(  
 path = "../part1\_shared\_files/inputs/grym\_parameter\_scenarios.xlsx",   
 sheet = "rec\_proportion\_scenarios"  
) |>  
 select(-ref)  
  
# Maturity at length scenarios  
maturity\_scenarios <- read\_xlsx(  
 path = "../part1\_shared\_files/inputs/grym\_parameter\_scenarios.xlsx",   
 sheet = "maturity\_ogive\_scenarios"  
) |>  
 select(-ref)  
  
# fixed parameters  
fixed\_pars <- read\_xlsx(  
 path = "../part1\_shared\_files/inputs/grym\_parameter\_scenarios.xlsx",   
 sheet = "fixed\_parameters")  
  
# load draws of M and CV\_R for each PR scenario  
pr\_cvR\_M\_draws\_scen <- read\_rds("../part1\_shared\_files/outputs/cvR\_M\_draws\_scen.rds")

The four PR scenarios are displayed in **?@tbl-pr-scens**. Parameter values for each of the maturity ogive curves are displayed in **?@tbl-mat-scens**, and fixed Grym input parameters are presented in **?@tbl-fixed-pars**.

maturity\_scenarios |>  
 flextable() |>  
 set\_header\_labels(  
 mat\_scen\_id = "Maturity ID",  
 source = "Source",  
 mat50Min = "Min length 50% mature (mm)",  
 mat50Max = "Max length 50% mature (mm)",  
 matrange = "Ogive ramp width (mm)"  
 )

**Table** **:** Alternative maturity-at-length ogive parameters, from Maschette et al (2021)

| **Maturity ID** | **Source** | **Min length 50% mature (mm)** | **Max length 50% mature (mm)** | **Ogive ramp width (mm)** |
| --- | --- | --- | --- | --- |
| mat-2010 | WG-EMM-2010 | 32.0 | 37.0 | 6.0 |
| mat-2021 | US-AMLR summer surveys | 37.6 | 44.3 | 8.8 |

# cit\_keys <- fixed\_pars |>  
# distinct(citation\_key) |>  
# drop\_na() |>  
# mutate(ref\_symb = tolower(as.roman(1:n())))  
  
fixed\_pars |>  
 select(-c(parameter\_tag, reference)) |>  
 flextable() |>  
 set\_header\_labels(  
 category = "Category",  
 parameter\_label = "Parameter",  
 value = "Value",  
 citation\_key = "Reference"  
 ) |>  
 merge\_v(j = 1) |>  
 flextable::fix\_border\_issues() |>  
 colformat\_md()

**Table** **:** Grym input parameters with fixed values accross simulation scenarios

| **Category** | **Parameter** | **Value** | **Reference** |
| --- | --- | --- | --- |
| PR model | Reference age | 2 | Maschette et al. (2021) |
| Recruitment distribution | Log-Normal |  |
| Mortality | Within-year natural mortality pattern | constant |  |
| Population model structure | First Age Class | 1 | Thanassekos et al. (2021) |
| Last Age Class | 7 | Constable and de la Mare (1996) |
| Age plus group | No |  |
| Time steps (days) | 365 |  |
| Length-at-age (von Bertalanffy) | t0 | 0 | Constable and de la Mare (1996) |
| L∞ (mm) | 60 | Constable and de la Mare (1996) |
| K | 0.48 | Thanassekos et al. (2021) |
| Start growth period (day/month) | 21/Oct | Thanassekos et al. (2021) |
| End growth period (day/month) | 12/Feb | Thanassekos et al. (2021) |
| Weight-at-length | Weight-length parameter - A (g) | 2.24E-06 | SC-CAMLR (2000) |
| Weight-length parameter - B | 3.314 | SC-CAMLR (2000) |
| Spawning season | Start of spawning season (day/month) | 15/Dec | Kawaguchi (2016) |
| End of spawning season (day/month) | 15/Feb | Kawaguchi (2016) |
| Selectivity ogive | Min length, 50% Selected (mm) | 30 | Thanassekos et al. (2021) |
| Max length, 50% Selected (mm) | 35 | Thanassekos et al. (2021) |
| Ramp width (mm) | 11 | Thanassekos et al. (2021) |
| Fishing effort | Start of Fishing Season (day/month) | 01/Dec | Thanassekos et al. (2021) |
| End of Fishing Season (day/month) | 30/Nov | Thanassekos et al. (2021) |
| B0 estimation | Uncertainty in B0 estimate | 0.361 | Kinzey (2021) |
| Start of monitoring interval (day/month) | 01/Jan | Thanassekos et al. (2021) |
| End of monitoring interval (day/month) | 15/Jan | Thanassekos et al. (2021) |
| Miscellaneous | Reference Date (day/month) | 01/Oct | Thanassekos et al. (2021) |
| Simulation options | Reasonable upper bound for Annual F | 1.5 | Constable and de la Mare (1996) |
| Projection length (number of years) | 20 |  |
| Number of iterations | 10001 |  |

# flextable::footnote(  
 # j = ~ value,   
 # i = ~ citation\_key %in% cit\_keys$citation\_key,   
 # ref\_symbols = left\_join(fixed\_pars, cit\_keys, by = "citation\_key") |> drop\_na(citation\_key) |> pull(ref\_symb),  
 # value = as\_paragraph\_md(  
 # fixed\_pars |> drop\_na(citation\_key) |> distinct(citation\_key) |> pull(citation\_key)  
 # ), inline = TRUE  
 # )

Next we create a look-up table specifying the simulation scenarios, determined by combining the PR scenarios and the two maturity ogive alternatives.

# Key of scenarios under consideration for Grym simulations  
grym\_scenarios\_key <- expand\_grid(  
 pr\_scen\_id = prop\_rec\_scenarios$pr\_scen\_id,  
 mat\_scen\_id = maturity\_scenarios$mat\_scen\_id  
) |>  
 mutate(scenario\_id = glue::glue("scn-{1:n()}"), .before = 1)  
  
# save for later  
write\_rds(grym\_scenarios\_key, "../part1\_shared\_files/inputs/scenarios\_key.rds")  
  
grym\_scenarios\_key |>  
 flextable() |>  
 set\_table\_properties(width = 0.6, layout = "autofit") |>  
 set\_header\_labels(  
 scenario\_id = "Scenario ID",  
 pr\_scen\_id = "PR Scenario ID",  
 mat\_scen\_id = "Maturity ID"  
 )

**Table** **:** Specification of parameter input scenarios

| **Scenario ID** | **PR Scenario ID** | **Maturity ID** |
| --- | --- | --- |
| scn-1 | PR-emm21 | mat-2010 |
| scn-2 | PR-emm21 | mat-2021 |
| scn-3 | PR-amlr | mat-2010 |
| scn-4 | PR-amlr | mat-2021 |
| scn-5 | PR-atlantida | mat-2010 |
| scn-6 | PR-atlantida | mat-2021 |
| scn-7 | PR-amlr-haul | mat-2010 |
| scn-8 | PR-amlr-haul | mat-2021 |

### 2.2.2 Set values under consideration

We also need to define the range of harvest rates over which population projections will be run. We chose to use equally spaced rates, ranging from no harvest () to a constant annual harvest rate of 25% of the pre-exploitation biomass (), with 0.25% increments between them.

# sequence of gamma values  
# i.e. testing fixed annual harvest rates from 0% to 25% of B0, for increments of 0.25%  
gamma\_seq <- seq(0, 0.25, by = 0.0025)  
gamma\_seq

[1] 0.0000 0.0025 0.0050 0.0075 0.0100 0.0125 0.0150 0.0175 0.0200 0.0225  
 [11] 0.0250 0.0275 0.0300 0.0325 0.0350 0.0375 0.0400 0.0425 0.0450 0.0475  
 [21] 0.0500 0.0525 0.0550 0.0575 0.0600 0.0625 0.0650 0.0675 0.0700 0.0725  
 [31] 0.0750 0.0775 0.0800 0.0825 0.0850 0.0875 0.0900 0.0925 0.0950 0.0975  
 [41] 0.1000 0.1025 0.1050 0.1075 0.1100 0.1125 0.1150 0.1175 0.1200 0.1225  
 [51] 0.1250 0.1275 0.1300 0.1325 0.1350 0.1375 0.1400 0.1425 0.1450 0.1475  
 [61] 0.1500 0.1525 0.1550 0.1575 0.1600 0.1625 0.1650 0.1675 0.1700 0.1725  
 [71] 0.1750 0.1775 0.1800 0.1825 0.1850 0.1875 0.1900 0.1925 0.1950 0.1975  
 [81] 0.2000 0.2025 0.2050 0.2075 0.2100 0.2125 0.2150 0.2175 0.2200 0.2225  
 [91] 0.2250 0.2275 0.2300 0.2325 0.2350 0.2375 0.2400 0.2425 0.2450 0.2475  
[101] 0.2500

Finally we generate a grid table with input parameter setups under each scenario, which will be the reference table for the Grym simulations.

# spread fixed parameters over columns  
fixed\_pars\_wide <- fixed\_pars |>  
 select(parameter\_tag, value) |>  
 pivot\_wider(names\_from = parameter\_tag)  
  
# merge fixed parameters with scenario's grid table  
grym\_scen\_inputs <- grym\_scenarios\_key |>  
 left\_join(maturity\_scenarios, by = "mat\_scen\_id") |>  
 select(-source) |>  
 add\_column(fixed\_pars\_wide)  
  
# tibble with list-columns for parameters requiring non-scalar objects  
grym\_scen\_setups <- grym\_scen\_inputs |>  
 mutate(across(.cols = c(pr\_ref\_age, first\_age, last\_age, nsteps, n.years, n\_iter), as.integer)) |>  
 mutate(across(.cols = c(t0:K, a, b, sel50Min:selrange, B0logsd, Fmax), as.numeric)) |>  
 rowwise() |>  
 mutate(  
 .keep = "unused",  
 Ages = list(first\_age:last\_age), # Age-classes  
 spawnI = list(get\_daily\_steps(spawning\_start, spawning\_end, ref\_date)),   
 monitorI = list(get\_daily\_steps(monitoring\_start, monitoring\_end, ref\_date)),  
 fishingI = list(get\_daily\_steps(fishing\_start, fishing\_end, ref\_date)),  
 f0 = days\_since\_ref(growth\_start, ref\_date)/365, # fraction of year at which growth starts,  
 f1 = days\_since\_ref(growth\_end, ref\_date)/365, # fraction of year at which growth ends  
 ) |>  
 mutate(  
 # draws of PR estimates (M, CV\_R, mnQ, vrQ)  
 prRecruitPars = pr\_cvR\_M\_draws\_scen[pr\_scen\_id],   
 # recruitment generating function for assumed rec dist  
 prRecruit = case\_when(   
 pr\_dist == "Log-Normal" ~ list(recLogNormal),  
 pr\_dist == "Gamma" ~ list(recGamma),  
 pr\_dist == "Invserse-Beta" ~ list(recInverseBeta)  
 ),  
 # Within-year M pattern  
 ms = case\_when(  
 M\_within\_year == "constant" ~ list(matrix(1, nsteps+1, length(Ages)))  
 ),   
 # sequence of harvest rates over which to run population projections  
 gamma = list(gamma\_seq),  
 outputs\_path = outputs\_path  
 )  
  
write\_rds(grym\_scen\_setups, "../part1\_shared\_files/inputs/grym\_scen\_setups.rds")  
  
grym\_scen\_setups

# A tibble: 8 × 34  
# Rowwise:   
 scenario\_id pr\_scen\_id mat\_scen\_id mat50Min mat50Max matrange pr\_ref\_age  
 <glue> <chr> <chr> <dbl> <dbl> <dbl> <int>  
1 scn-1 PR-emm21 mat-2010 32 37 6 2  
2 scn-2 PR-emm21 mat-2021 37.6 44.3 8.8 2  
3 scn-3 PR-amlr mat-2010 32 37 6 2  
4 scn-4 PR-amlr mat-2021 37.6 44.3 8.8 2  
5 scn-5 PR-atlantida mat-2010 32 37 6 2  
6 scn-6 PR-atlantida mat-2021 37.6 44.3 8.8 2  
7 scn-7 PR-amlr-haul mat-2010 32 37 6 2  
8 scn-8 PR-amlr-haul mat-2021 37.6 44.3 8.8 2  
# ℹ 27 more variables: pr\_dist <chr>, M\_within\_year <chr>, age\_plus <chr>,  
# nsteps <int>, t0 <dbl>, Linf <dbl>, K <dbl>, a <dbl>, b <dbl>,  
# sel50Min <dbl>, sel50Max <dbl>, selrange <dbl>, B0logsd <dbl>, Fmax <dbl>,  
# n.years <int>, n\_iter <int>, Ages <list>, spawnI <list>, monitorI <list>,  
# fishingI <list>, f0 <dbl>, f1 <dbl>, prRecruitPars <named list>,  
# prRecruit <list>, ms <list>, gamma <list>, outputs\_path <chr>

### 2.2.3 Run Grym models

In this section we run the Grym simulations, and thus the estimation of , for each scenario. We do this by sequentially mapping the function [run\_grym\_krill()](#sec-code-proj-wrapper-fnct) to each scenarios setup. Function run\_grym\_krill() is a wrapper that runs simulations within each scenario in parallel, using the projection function [KrillProjection()](#sec-code-proj-fnct). It then selects the and values for the scenario being evaluated.

# Note: runtime is substantial (~8hrs across 20 cores), so we don't want to run  
# this chunk on rendering!  
  
# Progress bar configuration  
handlers(handler\_progress(  
 format = ":spin :current/:total [:bar] :percent in :elapsed ETA: :eta",  
 width = 70)  
)  
  
# set-up cores for parallization  
future::plan(multisession, workers = future::availableCores()-1)  
  
# run simulations - needs to be wrapped in `with-progress` to get progress bars   
# for projection runs  
tictoc::tic()  
grym\_scen\_outputs <- progressr::with\_progress(  
 pmap(grym\_scen\_setups, run\_grym\_krill),   
 delay\_stdout = FALSE  
)  
tictoc::toc()  
  
# switch back to sequential computations  
future::plan(sequential)  
  
names(grym\_scen\_outputs) <- grym\_scen\_setups$scenario\_id  
  
# Write out object with all the results (compressing due to large size)  
write\_rds(grym\_scen\_outputs,  
 "../part1\_shared\_files/outputs/grym/grym\_scen\_results.rds",  
 compress = "gz",  
 compression = 9)

## 2.3 Results

### 2.3.1 Simulated Spawning Stock trajectories

[Figure 2.1](#fig-grym-sss-traject) presents the simulated trajectories of spawning Stock Status (), expressing the proportion of relative to , at each year of the projecting period. As expected, in all scenarios, trajectories show an increasing proportion of simulations in which drops bellow 20% of as the level of harvesting increases.

It is also clear that input scenarios with higher levels of recruitment variability and uncertainty in estimates (scen-3 to scen-8, check **?@tbl-sim-scenarios** and [Figure 1.1](#fig-M-RCV-draws) for reference) produce widely variable trajectories. As a result, a large proportion of simulations falls below the depletion threshold even in the absence of fishing (i.e. ).

grym\_scen\_outputs\_sub <- read\_rds("../part1\_shared\_files/outputs/grym/grym\_scen\_outputs\_sub.rds")  
  
# plots of simulated SSS across years, for each scenario under a subset of gammas  
# SSS trajectories that go below 0.2 are signaled in green  
p <- grym\_scen\_outputs\_sub |>  
 ggplot(aes(x = Year, y = SSS, group = Run)) +  
 geom\_path(  
 data = ~filter(.x, below\_dpl == FALSE),   
 alpha = 0.8, color = "gray75", linewidth = 0.3  
 ) +  
 geom\_path(  
 data = ~filter(.x, below\_dpl == TRUE),   
 alpha = 0.8, color = "#1B5E20", linewidth = 0.3  
 ) +  
 geom\_hline(yintercept = c(0.2), linetype = "dashed") +  
 scale\_y\_sqrt() +  
 guides(colour="none") +  
 labs(y = expression(paste("Spawning stock status (SSB/", SSB[0], ")"))) +  
 facet\_grid(rows = vars(scenario\_id), cols = vars(Gamma), scales = "free\_y")  
  
ggsave(  
 plot = p,   
 filename = "../part1\_shared\_files/outputs/grym/grym\_SSS\_scenarios.png",   
 width = 8,   
 height = 9  
)

|  |
| --- |
| Figure 2.1: Simulated trajectories of spawning stock status over time, for each scenario (rows) under a subset of the considered values (columns). Dashed line indicates the depletion threshold (0.2). Trajectories that fall below the 0.2 depletion threshold are signaled in green. |

The impact of increasing values on the biomass of spawners at the end of the projection period () is also evident in [Figure 2.2](#fig-grym-ssb0-ssb-dist-final-year). As expected, distributions of fall further from distributions as annual harvest rates increase.

In addition, plots in [Figure 2.2](#fig-grym-ssb0-ssb-dist-final-year) illustrate how variability in recruitment () and affect the escapement levels on the stock. Specifically, simulations under scenarios scn-3 to scn-8, which are based on estimates of and with higher variability (as shown by wider quantile intervals), tend to produce median values that drop below 75% of the median at lower values compared to those simulated under scenarios scn-1 and scn-2 (which have narrower quantile intervals).

# get simulated values of SSB0 and SSB in final year of the projection  
spawners <- grym\_scen\_outputs\_sub |>  
 filter(Year %in% max(Year)) |>  
 select(scenario\_id, Run, Gamma, SSB, SSB0) |>  
 pivot\_longer(cols = c(SSB , SSB0), names\_to = "metric") |>  
 mutate(  
 metric = if\_else(metric == "SSB", "SSBY", metric),  
 Year = if\_else(metric == "SSBY", "20", "0")  
 )  
  
# compute medians SSB0 and SSB across all simulations , under each scenario  
med\_spawners <- spawners |>  
 group\_by(scenario\_id, metric, Year, Gamma) |>  
 summarise(medians = median(value), .groups = "drop")   
  
  
# Compute escapement threshold value, i.e. 75% of median SSB0)  
# Note: values are constant across gammas, as simulated SSB0s are held constant  
# across gamma-specific projections.  
esc\_thresh <- med\_spawners |>  
 filter(metric == "SSB0") |>  
 mutate(esc\_thrs = medians\*0.75)  
  
p2 <- spawners |>  
 ggplot(aes(x = Year, y = value)) +  
 ggdist::stat\_interval() +  
 geom\_point(data = med\_spawners, aes(y = medians)) +  
 geom\_hline(data = esc\_thresh, aes(yintercept = esc\_thrs), linetype = "dashed") +  
 facet\_grid(scenario\_id ~ Gamma, scales = "free") +  
 labs(y = "Spawning Biomass", ) +  
 # scale\_colour\_grey(end = 0.2, start = 0.8)  
 scale\_color\_brewer(palette = "Greens", name = "Quantile Interval (prob)") +  
 theme(legend.position="bottom")  
  
ggsave(  
 plot = p2,   
 filename = "../part1\_shared\_files/outputs/grym/grym\_SSB0\_SSBY\_dstbn\_scenarios.png",   
 width = 8,   
 height = 9  
)

|  |
| --- |
| Figure 2.2: Quantile intervals (blue bars) and medians (dots) of simulated pre-explotation spawining biomass (Year 0) and spawning biomass at the final year of the projection (Year 20), across a subset of s (columns) for each scenario (rows). Dashed lines indicate the critical escapement threshold (i.e. 75% of median SSB0). |

### 2.3.2 Depletion probability and Escapement levels versus values

[Figure 2.3](#fig-grym-dplt-esc-gammas) shows the depletion probability[[3]](#footnote-69) and the escapement level[[4]](#footnote-70) at the considered range of harvest rates. Here are the main observations:

* For scenarios scn-3, scn-4, scn-7 and scn-8, probability of depletion would be considerably above the 10% critical limit even in the absence of fishing (i.e. ).
* In the remaining scenarios, depletion probability would remain under the 10% critical limit for values up to 0.075 (scn-6) and 0.125 (scn-1).
* Escapement levels stay above the 75% threshold for a large range of values, depending on the input scenario. For example, simulations under scn-1 inputs suggest that stock would remain above the 75% escapement threshold for constant harvest rates up to 10% of (i.e. ), while for scn-8 the stock would fall below precautionary escapement levels under constant harvest rates above 2% of (i.e. ).
* Both plots illustrate how alternative maturity ogives can have a significant impact on the simulation outcomes, as highlighted by the separation between pairs of scenarios (scn-1 & scn-2, scn-3 & scn-4, etc.). Scenarios using the mat-2010 maturity ogive, such as scn-1, scn-3, scn-5, and scn-7, which yields a higher percentage of mature individuals at smaller sizes, show greater resilience to fishing pressure. The projections suggest that under these scenarios, the stock could withstand higher levels of before falling to unsustainable levels of depletion risk and escapement.

grym\_dpl\_esc\_gammas <- read\_rds("../part1\_shared\_files/outputs/grym/grym\_dpl\_esc\_gammas.rds")  
  
# Plot for depletion probability under considered gammas  
p\_dpl <- plot\_gammas\_vs\_rule(  
 dt = grym\_dpl\_esc\_gammas,   
 gamma = Gamma,   
 rule\_value = Pr\_depleted,   
 scen = scenario\_id,  
 thresh = 0.1,   
 ylab = "Pr[min(SSB/SSB0) < 0.2] (Years 1 - 20)",   
 xlab = expression(gamma),  
 title = "Depletion",   
 scen\_label = "Scenario ID"  
 )  
  
# Plot for escapement level under considered gammas  
p\_esc <- plot\_gammas\_vs\_rule(  
 dt = grym\_dpl\_esc\_gammas,   
 gamma = Gamma,   
 rule\_value = Escapement,   
 scen = scenario\_id,   
 thresh = 0.75,  
 title = "Escapement",   
 ylab = "med(SSB)/med(SSB0) in Final Year",  
 xlab = expression(gamma),  
 scen\_label = "Scenario ID"  
 )  
  
p\_dpl/p\_esc + plot\_layout(guides = 'collect')

|  |
| --- |
| Figure 2.3: Depletion probabilities and escapement levels at considered harvest rates under each scenario. Horizontal dashed lines indicate the 10% probability limit of the depletion rule (top plot) and the 75% critical threshold of the escapement rule (bottom plot). |

### 2.3.3 Estimated under each input scenario

Finally, we determine the precautionary harvest rate for each input scenario based on the 3-stage decision rule (**?@tbl-grym-selected-gamma**).

As expected, the chosen Proportional Recruitment scenario had the largest impact on the estimated values of . Specifically, scenarios using PR estimates reflecting lower variability and uncertainty in and (scn-1 and scn-2) resulted in higher estimates of , based on the escapement rule (). Conversely, scenarios scn-3, scn-4, scn-7 and scn-8, based on PR estimates conveying the highest variability in and (PR-amlr and PR-amlr-haul), returned estimates of 0 based on the depletion rule (). In fact, as seen in [Figure 2.3](#fig-grym-dplt-esc-gammas), under these scenarios, the risk of depletion in the absence of fishing is already very high, between approximately 25% and 45%.

The effect of maturity-at-length on estimates is also evident, as previously observed in [Figure 2.3](#fig-grym-dplt-esc-gammas). Scenarios employing the mat-2021 ogive yielded lower estimates of compared to scenarios using the mat-2010 ogive. Under the same selectivity ogive, the mat-2021 ogive allows a larger number of immature individuals to be vulnerable to the fishery than the mat-2010 ogive. This leads to a reduced incidence of spawners in an exploited population, causing a faster decline of from over the projection period. Consequently, the estimates of are lower and more conservative when the mat-2021 ogive is applied.

Scenarios scn-5 and scn-6 are also interesting because they demonstrate how variability in maturity ogives can affect which of the decision rules determines the choice of . Specifically, population simulations using the mat-2021 ogive (scn-6), which sample midpoints from a larger range of values (i.e. higher variability) than the mat-2010 ogive (scn-5), hit the depletion rule at lower values than the escapement rule (i.e. ).

grym\_gammas\_tbl <- read\_rds("../part1\_shared\_files/outputs/grym/grym\_gammas\_tbl.rds")  
  
# gammas in math format  
gamma\_symb <- c("\\gamma\_1", "\\gamma\_2", "\\gamma\_p")  
  
grym\_gammas\_tbl |>  
 mutate(across(-scenario\_id, as.numeric)) |>  
 mutate(across(c(Gamma\_1, Gamma\_2), ~if\_else(is.infinite(.), 0, .))) |>  
 left\_join(grym\_scenarios\_key, by = "scenario\_id") |>  
 relocate(Gamma\_1:Gamma\_choice, .after = last\_col()) |>  
 mutate(gamma\_p = pmin(Gamma\_1, Gamma\_2)) |>  
 select(-Gamma\_choice) |>  
 flextable() |>  
 set\_table\_properties(width = 0.8, layout = "autofit") |>  
 #bold(j = ~ gamma\_p) |>  
 bold(j = ~ Gamma\_1, i = ~ Gamma\_1 < Gamma\_2) |>  
 bold(j = ~ Gamma\_2, i = ~ Gamma\_2 < Gamma\_1) |>  
 vline(j = ~ Gamma\_2, border = fp\_border\_default(width = .5)) |>  
 bg(j = ~ gamma\_p, bg = "#E8F5E9", part = "all") |>  
 set\_header\_labels(  
 scenario\_id = "Scenario ID",  
 pr\_scen\_id = "PR Scenario ID",  
 mat\_scen\_id = "Maturity ID"  
 ) |>  
 width(j = ~ Gamma\_1 + Gamma\_2 + gamma\_p, width = 10) |>  
 compose(j = ~ Gamma\_1 + Gamma\_2 + gamma\_p,   
 part = "header",   
 value = as\_paragraph(as\_equation(gamma\_symb))) |>  
 align(j = ~ Gamma\_1 + Gamma\_2 + gamma\_p, align = "center")

**Table** **:** Estimated precautionary harvest rate under each Grym inputs scenario

| **Scenario ID** | **PR Scenario ID** | **Maturity ID** |  |  |  |
| --- | --- | --- | --- | --- | --- |
| scn-1 | PR-emm21 | mat-2010 | 0.1275 | **0.1075** | 0.1075 |
| scn-2 | PR-emm21 | mat-2021 | 0.1025 | **0.0900** | 0.0900 |
| scn-3 | PR-amlr | mat-2010 | **0.0000** | 0.0375 | 0.0000 |
| scn-4 | PR-amlr | mat-2021 | **0.0000** | 0.0300 | 0.0000 |
| scn-5 | PR-atlantida | mat-2010 | 0.0800 | **0.0775** | 0.0775 |
| scn-6 | PR-atlantida | mat-2021 | **0.0675** | 0.0700 | 0.0675 |
| scn-7 | PR-amlr-haul | mat-2010 | **0.0000** | 0.0200 | 0.0000 |
| scn-8 | PR-amlr-haul | mat-2021 | **0.0000** | 0.0150 | 0.0000 |

## 2.4 Wrapping Up

We now have estimates of using the current Grym base-case implementation for the Krill stock under alternative input parameter values. The next step is to replicate this estimation process using the openMSE/DLMtool and assess whether these two frameworks could be used interchangeably.

## 2.5 Supplementary Code

#### Wrapper function for running simulations across scenarios

#### Krill Base-case projection function

This is modified version of the KrillProjection() function available in CCAMLR’s Krill base-case [repo](https://github.com/ccamlr/Grym_Base_Case/blob/Simulations/3_Code/Source/Projection_function.R) that adds arguments ms and Fmax, for convenience.

# 3. openMSE-Grym Approximation

Reproducing the Grym’s base case for the krill fishery using the openMSE framework

## 3.1 Introduction

In this section, we aim to configure and run the models under the {openMSE} package (Hordyk et al., 2021) to approximate Grym’s base-case implementation for the assessment and management of the Antarctic Krill fishery. Our primary objective for this analysis is to determine whether the openMSE framework can generate estimates of precautionary harvest rates () similar to those obtained under the Grym framework, for a range of alternative scenarios involving key stock parameters, as described in [Chapter 2](#sec-grym-sims). The ability to achieve comparable results will establish the suitability of the openMSE framework to model the population dynamics of the Krill stock, and subsequently, its applicability to evaluate prospective alternative management options for the fishery.

In its essence, like the Grym approach, the openMSE framework is based on an stochastic age-structured population model that simulates fishery dynamics while taking into account uncertainties about stock parameters and system dynamics. This enables the identification of potential management strategies that are robust to a range credible scenarios of the fishery system. Details about the openMSE’s components and features are available on its [documentation site](https://openmse.com/).

However, there are fundamental structural differences between the Grym and the openMSE frameworks that may affect the success of the approximation. In the following subsection we describe the main discrepancies between the two approaches in detail.

## 3.2 Inherent differences between openMSE and Grym

Here we list relevant built-in differences between the Grym’s base-case implementation and the openMSE framework in terms of population model structure.

### 3.2.1 Temporal Resolution

The openMSE’s population model describes a fished stock and its dynamics at yearly time-steps, whereas Grym represents populations at daily time-steps.

Grym’s daily-based structure allows to model stock characteristics at a day-level resolution, which permits the specification of whithin-year patterns in e.g. growth, spawning, natural and fishing mortality. It also provides the flexibility to evaluate population parameters and reference points at specific periods of the year (e.g. during the spawning season or fishing period).

On the other hand, the openMSE framework is constrained to yearly-level computations and hence it is unable to account for within-year features nor to derive the stock status at particular fractions of the year - reported annual stock quantities are strictly related to the stock status at start of each year.

A concrete example of deviations between the two approaches related to temporal resolution is the calculation of key biomass quantities. In the Grym base-case, yearly and are computed relative to the spawning period, and is computed for a specific monitoring period. By comparison, under the openMSE dynamics model, these quantities are purely related to the first day of the year.

### 3.2.2 Stock-Recruitment Relationship

In contrast with the Grym approach, the openMSE framework uses a conventional stock-recruitment model to compute the expected number of recruits given the spawning stock size.

As described in [Section 2.1.1](#sec-grym-framework), the simulation of recruitment in Grym is assumed to be independent of the stock size, except when the stock falls below a critical depletion point (20% of ) in the preceding year. In this scenario, the simulated recruitment is penalized by a reduction factor calculated as the percentage drop from the critical point.

Under the openMSE approach, yearly recruitments are generated as random deviates from a mean recruitment that is dictated by a standard stock-recruitment relationship (the Beverton-Holt or the Ricker curves), which specifies the expected number of recruits at any given stock size. However, unlike Grym, openMSE does not offer the option to penalize recruitment deviates based in depletion levels.

As a result, when the stock falls below critical depletion levels, the strength of simulated recruitment in the two frameworks will tend to diverge, especially under high levels of recruitment variability.

### 3.2.3 Historical Period

The openMSE tool takes into account for the historical period of the fishery leading up to a current state, before projecting the population forward under potential future management strategies under evaluation. In the Grym base-case implementation, stock projections start from a pre-exploitation state without considering information on historical trends or current status of the fishery.

An obvious way to align the two approaches would be to skip the historical period in the openMSE simulation. However, for the current version of the openMSE, and in particular its {MSEtool} package (vr packageVersion("MSEtool")), a minimum historical period of 5 years is required. Local changes to the {MSEtool} codebase allowed the reduction of the minimum required historical period to 2 years.

Therefore, even when an historical period with no fishing mortality is specified, the openMSE simulations will still have an additional 2-year lead-up period subject to stochasticity, which is not present in the Grym implementation.

### 3.2.4 Unfished Reference Points Calculation

The Grym approach includes recruitment deviations in the calculation of unfished/pre-exploitation reference points, such as and . The openMSE approach provides similar estimates, referred to as “dynamic unfished reference points”. Under both approaches, these quantities are used to evaluate the status of the stock at different time points of the projection period for each of the management strategies under consideration.

However, there is a key divergence between approaches regarding the estimation of . In the Grym approach, is estimated stochastically within each simulation by taking the median of multiple realizations of from repeated samples of initial age structures. In contrast, the openMSE approach computes based on one single random sample of initial age structure generated at each given simulation. Consequently, estimates of under the Grym approach will tend to be more stable (i.e., have a lower variance) across simulations than those calculated in the openMSE approach. This difference could be especially pronounced in situations of high variability in stock parameters, such as natural mortality and recruitment.

## 3.3 Configuring the openMSE

Here we describe how the openMSE components are configured to approximate the estimation of precautionary harvest rates for the krill fishery as it is performed under the Grym base-case implementation[[5]](#footnote-91).

The openMSE framework requires the specification of 3 main components to run a Management Strategy Evaluation (MSE):

1. **Operating Model** ([OM](https://openmse.com/object-operating-model/)): containing parameters specifying the characteristics of the population and fishery dynamics, as well as parameters required to simulate the collection of data and the implementation of management procedures.
2. **Management Procedure(s)** ([MP](https://openmse.com/features-management-procedures/)): A MP defines a set of rules specifying how the fishery will be managed during the projection period (e.g. setting a fixed annual total allowable catch, TAC).
3. **Performance Metric(s)** ([PM](https://openmse.com/features-calculating-performance/)): A PM summarises the MSE outputs in order to evaluate the performance of the considered MPs against the management objectives specified for the fisheries, as e.g. the expected level of depletion over the projection period associated with a given MP.

We start off by building OMs for the 8 inputs scenarios under consideration.

### 3.3.1 Build OMs for input scenarios

As described in [Section 2.2.1](#sec-grym-input_scens), input scenarios forming alternative parameter setups for modelling the Krill fishery were generated from combining 4 different PR scenarios with 2 alternative maturity ogive curves. In order to run MSE simulations for each of the considered scenarios, we need to build scenario-specific OMs.

Therefore, OM parameter values were selected to closely approximate the parameter setups used in the Grym analyses presented in [Chapter 2](#sec-grym-sims). Specifically, for each scenario-specific OM, all OM parameters were held constant except for:

* Natural Mortality **M**, provided as random draws from the fitted PR model associated with the scenario (generated in [Chapter 1](#sec-generate-RCV-M));
* Recruitment Process error **Perr**, derived from random draws of , also from the fitted PR model;
* maturity-at-length parameters **L50** and **L50\_95** associated with the scenario

The function [build\_OM\_grym\_approx()](#sec-openmse-om-builder) was created to build OMs for the parameters values specified under each input scenario, as follows.

openmse\_scen\_OMs <- grym\_setups |>  
 select(scenario\_id:matrange, t0:Ages, prRecruitPars) |>  
 rowwise() |>  
 mutate(  
 OM = list(  
 build\_OM\_grym\_approx(  
 om\_name = glue::glue("krill\_grym\_approx\_{scenario\_id}"),   
 maxage = last(Ages), Linf = Linf, K = K, t0 = t0,   
 mat50Min = mat50Min, mat50Max = mat50Max, matrange = matrange,   
 sel50Min = sel50Min, sel50Max = sel50Max, selrange = selrange,   
 a = a, b = b,   
 M\_draws = pull(prRecruitPars, M) , RCV\_draws = pull(prRecruitPars, CV),  
 maxF = Fmax, n\_iter = n\_iter, proj\_yrs = n.years  
 )  
 )  
 ) |>  
 pull(OM)  
  
names(openmse\_scen\_OMs) <- grym\_setups$scenario\_id  
  
write\_rds(openmse\_scen\_OMs, "../part1\_shared\_files/inputs/openmse\_scen\_OMs.rds")

The openMSE framework includes a report generation tool that was used to document in detail the choice of OM parameter values for one of the input scenarios (*scn-1*), for reference.

# initialize OM report for scn-1  
OMinit(name = "OM\_krill\_grym\_approx\_scn-1", files = "rmd", overwrite = FALSE)

The documentation step is performed externally to this document. Once the OM is appropriately documented, we compile the report. The resultant document can be viewed in [Chapter 4](#sec-annex-1).

|  |
| --- |
| Important |
| Due to a bug in MSEtool::OMdoc(), the number of years to project the population forward must be set to 50 years, otherwise the markdown rendering process fails. This issue is exclusive to the OM\_doc function. |

# changed version of the MSEtool::OMdoc() that solves issues with the inclusion  
# of bibliographic references, provides better integration in another document  
# as an embedded html. Also adds option to change html theme  
source("OMdoc\_dmp.r")  
  
# Length of projection must be 50 years, otherwise compilation crashes due to  
# bug in OMdoc  
OM\_to\_report <- openmse\_scen\_OMs$`scn-1`  
OM\_to\_report@proyears <- 50  
  
OMdoc\_dmp(  
 OM = OM\_to\_report,   
 openFile = FALSE,  
 rmd.source = "OM\_krill\_grym\_approx\_scn-1.rmd",   
 bib\_file = "../../references.json",  
 html\_theme = "lumen")

Relevant choices to approximate the OMs specification to the Grym setups include:

* Set up the historical period to two years (minimum restricted by the basecode).
* Stock assumed to remain unexploited during the historical period.
* Define a range of high values (0.9 - 0.95) for the steepness of the Beverton-Holt stock-recruitment model in order to simulate yearly recruitments that are weakly dependent on stock size, unless when the stock is below 20% of its pre-exploitation size.
* Life-history parameters defining individual growth (i.e. **Linf**, **K**, **t0**, **a** and **b**) assumed to be known without error.
* Management procedures assumed to be perfectly implemented, i.e. annual catches never exceed nor fall short of the stipulated TACs.

### 3.3.2 Define Management Plans

The openMSE framework define Management Plans (MPs) as programmatic functions establishing how the fishery could prospectively be managed, by running a set of calculations and criteria on fishery information and returning management recommendations.

In the context of the openMSE-Grym approximation, and Krill’s current management approach, we want to specify one MP for each of the considered harvest rates . Thus, each MP establishes a constant harvest rate policy with a recommended fixed annual catch limit of throughout the projection period.

As explained in [Section 2.1.1](#sec-grym-framework), represents an estimate of pre-exploitation biomass which, similarly to the Grym approach, is simulated here as

where the standard deviation , conveying the observational error in survey estimates of , is estimated externally from survey data (e.g. Kinzey, 2021). The -based MPs are set using the estimated dynamic unfished stock biomass (i.e. including recruitment process error) at the end of the historical period, serving as a proxy for the pre-exploitation biomass . This is to approximate the Grym implementation, which incorporates recruitment variability in the estimation of reference points.

Below are the custom functions we developed to generate the -based MPs. These functions need to be defined as objects of class MP in order to be used with openMSE’s simulation functions.

### 3.3.3 Set Performance Metrics

Lastly, we must define Performance Metrics (PMs) to evaluate the -based MPs under consideration. In the context of Krill management, and as seen in [Chapter 2](#sec-grym-sims), there are two PMs of interest:

* the depletion probability, the probability of being below 20% of its pre-exploitation level at any year of a 20-year projection period)
* Escapement level, the median of simulated estimates at the end of the projection period relative to the median of simulated estimates

Following the same approach taken in the definition of MPs and in alignment with the Grym implementation, both PM calculations use the estimate of unfished spawning biomass at the end of the historical period as the pre-exploitation spawning biomass .

We create two functions, PD() and ESC(), of class PM to formalize these metrics for use within the openMSE package.

## 3.4 Run MSE Simulations

With all the necessary components specified for the openMSE framework, we are now ready to run the MSE simulations for each of the scenario-specific OMs defined above.

Due to the large number of simulations (10k) and the extensive quantity of -based MPs involved in each MSE run, we use the “slice-apply-combine” parallelization feature available on the main function MSEtool::runMSE() to speed up computations.

# Note: runtime is substantial (~2.5 days across 20 cores), so we don't want to run  
# this chunk on rendering!  
  
openmse\_scen\_OMs |>  
 imap(\(x, y){  
  
 cli::cli\_h1("Starting MSE run for {y} @ {Sys.time()}")  
   
 # Simulate observational errors in survey estimates of B0. `B0logsd` is the  
 # SD of survey estimates of (log) B0 conveying the observational error in  
 # surveys, assumed to be log-Normally distributed. `B0logsd` is calculated  
 # externally from survey data.  
 b0lgsd <- grym\_setups |> filter(scenario\_id == y) |> pull(B0logsd)  
 B0err\_draws <- rlnorm(x@nsim, -b0lgsd^2/2, b0lgsd)  
   
 # build MPs for considered gammas  
 gammas <- grym\_setups |> filter(scenario\_id == y) |> pull(gamma) |> pluck(1)  
 gamma\_B0\_MPs <- build\_gammaB0(  
 gammas = gammas,   
 nyears = x@nyears,  
 B0err = B0err\_draws)  
   
 # run MSE for current scenario   
 tictoc::tic()  
 mse\_output <- runMSE(OM = x, MPs = gamma\_B0\_MPs, parallel = "sac")  
 runtime <- tictoc::toc(quiet = TRUE)  
   
 # write out mse outputs  
 write\_rds(  
 mse\_output,  
 file = fs::path(model\_outputs\_path, glue::glue("openmse\_mse\_outputs\_{y}.rds")),   
 compress = "gz"  
 )  
   
 cli::cli\_alert\_success("Finished MSE for {y}: {runtime$callback\_msg}")  
 })

Next we extract the most relevant outputs from the MSE simulations, while also computing the precautionary harvest rate for each input scenario. As described in [Section 2.1](#sec-grym-intro), is derived based on the three-step decision rule involving the depletion probability and escapement performance metrics.

# ------------------------------------------------------------------------------  
# Utility function to extract relevant results and perform gammas selection  
# ------------------------------------------------------------------------------  
mse\_extract\_gammas <- function(mseObj\_file, SSB0\_type = "equilibrium"){  
  
 cli::cli\_alert("\n Processing file {mseObj\_file} @ {Sys.time()}")  
   
 mse\_out <- read\_rds(mseObj\_file)  
   
 # SSB0 data and source metrics functions, based on choice of type of reference point  
 if(SSB0\_type == "equilibrium"){  
 SSB0 <- mse\_out@OM$SSB0   
 source("part1\_openMSE\_GRYM\_approx/3\_openMSE\_sims/krill\_mngnt\_ccamlr\_metrics\_equilibrium.R")  
 }else   
 if(SSB0\_type == "dyn\_unfished"){  
 SSB0 <- mse\_out@RefPoint$Dynamic\_Unfished$SSB0[, mse\_out@nyears]  
 source("part1\_openMSE\_GRYM\_approx/3\_openMSE\_sims/krill\_mngnt\_ccamlr\_metrics\_dynamic.R")  
 }  
   
 # get considered gammas  
 gammas <- as.numeric(str\_replace(mse\_out@MPs, "gammaB0\_", ""))  
   
 # extract projected SSB and cast it into a dataframe  
 dimnames(mse\_out@SSB) <- list(sim = 1:mse\_out@nsim, gamma = gammas, Year = 1:mse\_out@proyears)  
 ssb\_proj <- reshape2::melt(mse\_out@SSB, value.name = "SSB") |> as\_tibble()  
 ssb0 <- tibble(  
 sim = 1:mse\_out@nsim,   
 SSB0 = SSB0  
 )  
   
 # merge in SSB0, calculate yearly spawning stock status (SSS), and identify  
 # simulations where SSS < 0.2 at any point in the time-series  
 ssb\_proj <- left\_join(ssb\_proj, ssb0, by = "sim") |>  
 group\_by(sim, gamma) |>  
 mutate(  
 SSS = SSB/SSB0,  
 below\_dpl = if\_else(min(SSS) < 0.2, TRUE, FALSE)  
 ) |>  
 ungroup()  
   
 # Compute depletion metric for each gamma  
 dep\_metric <- PD(mse\_out)  
   
 # Compute escapement metric for each gamma  
 esc\_metric <- ESC(mse\_out)  
   
 mse\_metrics <- tibble(gamma = gammas, PD = dep\_metric@Mean, ESC = esc\_metric@Mean)  
   
 # Derive gammas that satisfy the depletion and the escapement rules, and the  
 # final gamma\_p as min(gamma\_1, gamma\_2)  
 gamma\_results <- mse\_metrics |>  
 summarise(  
 gamma1 = max(max(gamma[PD <= 0.1]), 0),   
 gamma1\_approx = approx(PD, gamma, 0.1)$y,   
 gamma2 = max(gamma[ESC >= 0.75]),  
 gamma2\_approx = approx(ESC, gamma, 0.75)$y  
 ) |>  
 mutate(selected\_gamma = if\_else(gamma1 < gamma2, 1, 2))  
   
 # return results in tibble with list-columns  
 tibble(  
 ssb\_proj = list(ssb\_proj), dep\_metric = list(dep\_metric),   
 esc\_metric = list(esc\_metric), mse\_metrics = list(mse\_metrics),   
 gamma\_results = list(gamma\_results)  
 )  
}  
# -------------------------------------------------------------------------------  
  
  
# -- Results and gammas for unfished dynamics SSB0 and dynamic B0  
openmse\_2yrhst\_dynB0\_dynSSB0\_scen <- fs::dir\_ls(model\_outputs\_path, regexp = "scn") |>  
 map\_df(~mse\_extract\_gammas(.x, SSB0\_type = "dyn\_unfished"), .id = "fname") |>  
 mutate(scen\_id = str\_extract(fname, "scn-\\d+"), .before = 1, .keep = "unused")  
  
# -- write out selected gammas  
openmse\_2yrhst\_dynB0\_dynSSB0\_scen |>  
 select(scen\_id, gamma\_results) |>  
 unnest(gamma\_results) |>  
 select(-contains("approx")) |>  
 write\_rds(  
 file = fs::path(results\_path, "openmse\_2yrhst\_dynB0\_dynSSB0\_scen\_gamma\_select.rds")  
 )  
  
# -- write out performance metrics  
openmse\_2yrhst\_dynB0\_dynSSB0\_scen |>  
 select(scen\_id, mse\_metrics) |>  
 unnest(mse\_metrics) |>  
 write\_rds(file = fs::path(results\_path, "openmse\_2yrhst\_dynB0\_dynSSB0\_scen\_metrics.rds"))  
  
# -- write out simulated SSB trajectories under all scenarios  
openmse\_2yrhst\_dynB0\_dynSSB0\_scen |>  
 select(scen\_id, ssb\_proj) |>  
 unnest(ssb\_proj) |>  
 write\_rds(file = fs::path(results\_path, "openmse\_2yrhst\_dynB0\_dynSSB0\_scen\_ssb.rds"),  
 compress = "gz"  
 )

## 3.5 Results

# ----------------------------  
# Read-in relevant outputs  
# ----------------------------  
# import scenario key  
scenarios\_key <- read\_rds("../part1\_shared\_files/inputs/scenarios\_key.rds") |>  
 rename(scen\_id = scenario\_id)  
  
# read in grym projections under each scenario, for a subset of gamma  
grym\_scen\_outputs\_sub <- read\_rds("../part1\_shared\_files/outputs/grym/grym\_scen\_outputs\_sub.rds")  
  
# subset of gammas to display in plots  
gamma\_subset <- unique(grym\_scen\_outputs\_sub$Gamma)  
  
# Grym gamma estimates  
grym\_gammas\_tbl <- read\_rds("../part1\_shared\_files/outputs/grym/grym\_gammas\_tbl.rds") |>  
 mutate(across(-scenario\_id, as.numeric)) |>  
 mutate(across(c(Gamma\_1, Gamma\_2), ~if\_else(is.infinite(.), 0, .))) |>  
 rename(scen\_id = scenario\_id, gamma1 = Gamma\_1, gamma2 = Gamma\_2, selected\_gamma = Gamma\_choice)  
  
# openmse   
openmse\_scen\_metrics <- read\_rds(  
 fs::path(results\_path, "openmse\_2yrhst\_dynB0\_dynSSB0\_scen\_metrics.rds")  
)  
  
# open mse gamma estimates  
openmse\_scen\_gamma\_select <- read\_rds(  
 fs::path(results\_path, "openmse\_2yrhst\_dynB0\_dynSSB0\_scen\_gamma\_select.rds")  
)

### 3.5.1 Spawning stock trajectories

Simulated trajectories of spawning Stock Status () across the projecting period are presented in [Figure 3.1](#fig-openmse-sss-traject). As expected, trajectories show an increasing proportion of simulations in which drops bellow 20% of as the level of harvesting increases. Consistent with the Grym analysis ([Figure 2.1](#fig-grym-sss-traject)), there is greater variability in the trajectories across simulations in scenarios scen-3 to scen-8, reflecting the higher variance in simulated recruitment and natural mortality in those scenarios. Consequently, a substantial portion of simulations show stock depletion levels falling below the critical 20% threshold at low levels of fishing pressure (including when fishing is not present, i.e., ).

# plots of simulated SSS across years, for each scenario under a subset of gammas  
# SSS trajectories that go below 0.2 are signaled in green  
p <- openmse\_scen\_ssb |>  
 filter(gamma %in% gamma\_subset) |>  
 ggplot(aes(x = Year, y = SSS, group = sim)) +  
 geom\_path(  
 data = ~filter(.x, below\_dpl == FALSE),   
 alpha = 0.8,   
 color = "gray75",   
 linewidth = 0.3  
 ) +  
 geom\_path(  
 data = ~filter(.x, below\_dpl == TRUE),   
 alpha = 0.8,   
 color = frmwk\_colours["openMSE"],   
 linewidth = 0.3  
 ) +  
 geom\_hline(yintercept = c(0.2), linetype = "dashed") +  
 scale\_y\_sqrt() +  
 guides(colour="none") +  
 labs(y = expression(paste("Spawning stock status (SSB/", SSB[0], ")"))) +  
 facet\_grid(rows = vars(scen\_id), cols = vars(gamma), scales = "free\_y")  
  
  
ggsave(  
 plot = p,   
 filename = "../part1\_shared\_files/outputs/openmse/openmse\_fig\_SSS\_trajectories\_gammas\_scens.png",   
 width = 8,   
 height = 9  
)

|  |
| --- |
| Figure 3.1: Simulated trajectories of spawning stock status over time, for each scenario (rows) under a subset of the considered values (columns). Dashed line indicates the depletion threshold (0.2). Trajectories that fall below the 0.2 depletion threshold are signaled in blue. |

[Figure 3.2](#fig-openmse-ssb0-ssb-dist-final-year) compares the distributions of simulated spawning stock biomass at a two time points: before exploitation (Year 0, and after a 20-year period of harvesting (). The comparison is shown for a selection of annual harvest rates considered in the analysis. As expected, the decline of relative to becomes more pronounced as the value of increases.

In line with the Grym analysis, plots in [Figure 3.2](#fig-openmse-ssb0-ssb-dist-final-year) also highlight the effects of high variability in recruitment and natural mortality in projected stock escapement levels. Simulations conducted under scenarios scn-3 to scn-8, which are based on Proportional Recruitment estimates yielding higher variance in simulated values of and , return median estimates that are below 75% of the median at lower values, in contrast to simulations conducted under scenarios scn-1 and scn-2.

# get simulated values of SSB0 and SSB in final year of the projection  
spawners <- openmse\_scen\_ssb |>  
 filter(  
 gamma %in% gamma\_subset,  
 Year %in% max(Year)  
 ) |>  
 select(scen\_id, sim, gamma, SSB, SSB0) |>  
 pivot\_longer(cols = c(SSB , SSB0), names\_to = "metric") |>  
 mutate(  
 metric = if\_else(metric == "SSB", "SSBY", metric),  
 Year = if\_else(metric == "SSBY", "20", "0")  
 )  
  
# compute medians SSB0 and SSB across all simulations , under each scenario  
med\_spawners <- spawners |>  
 group\_by(scen\_id, metric, Year, gamma) |>  
 summarise(medians = median(value), .groups = "drop")   
  
# Compute escapement threshold value, i.e. 75% of median SSB0)  
# Note: values are constant across gammas, as simulated SSB0s are held constant  
# across gamma-specific projections.  
esc\_thresh <- med\_spawners |>  
 filter(metric == "SSB0") |>  
 mutate(esc\_thrs = medians\*0.75)  
  
p2 <- spawners |>  
 ggplot(aes(x = Year, y = value)) +  
 ggdist::stat\_interval() +  
 geom\_point(data = med\_spawners, aes(y = medians)) +  
 geom\_hline(data = esc\_thresh, aes(yintercept = esc\_thrs), linetype = "dashed") +  
 facet\_grid(scen\_id ~ gamma, scales = "free") +  
 labs(y = "Spawning Biomass" ) +  
 scale\_color\_brewer(palette = "Blues", name = "Quantile Interval (prob)") +  
 theme(legend.position="bottom")  
  
ggsave(  
 plot = p2,   
 filename = "../part1\_shared\_files/outputs/openmse/openmse\_SSB0\_SSBY\_dstbn\_scens.png",   
 width = 8,   
 height = 9  
)

|  |
| --- |
| Figure 3.2: Quantile intervals (blue bars) and medians (dots) of simulated pre-explotation spawining biomass (Year 0) and spawning biomass at the final year of the projection (Year 20), across a subset of values (columns) for each scenario (rows). Dashed lines indicate the critical escapement threshold (i.e. 75% of median SSB0). |

### 3.5.2 Depletion probability and Escapement levels versus values

An overall view of the potential impact of various fixed annual harvest rates on the health of the stock, measured in terms of depletion probability and escapement levels, is shown in [Figure 3.3](#fig-openmse-dplt-esc-gammas). Main findings follow broadly those found in the Grym analysis, e.g.:

* For scenarios scn-3, scn-4, scn-7 and scn-8, probability of depletion would be substantially above the 10% critical limit even in the absence of fishing (i.e. ). In the remaining scenarios, depletion probability would remain under the 10% critical limit for values up to approximately 0.05 (scn-6) and 0.1 (scn-1).
* values at which the stock is maintained at escapement levels above the critical 75% threshold also vary markedly between input scenarios. Scenario scn-8 simulates a stock with high vulnerability to fishing pressure, only able to sustain escapement levels above the critical point for harvest rates up to 1.5% of . On the other hand, under scenario scn-1, simulations suggest that the stock would remain above 75% escapement levels if the harvest rates are kept at least below 8% of .
* Plots also show the effect of using alternative maturity ogives in the simulations, as evidenced by the separation between the pairs of scenarios scn-1 & scn-2, scn-3 & scn-4, etc. In comparison with scenarios under the maturity ogive mat-2021, those using the mat-2010 maturity ogive (scn-1, scn-3, scn-5, and scn-7) would cope with higher levels of before dropping into unsustainable levels of depletion risk and escapement.

openmse\_scen\_metrics <- read\_rds("../part1\_shared\_files/outputs/openmse/openmse\_scen\_metrics.rds")  
  
# Plot for depletion probability under considered gammas  
p\_dpl <- plot\_gammas\_vs\_rule(  
 dt = openmse\_scen\_metrics,   
 gamma = gamma,   
 rule\_value = PD,   
 scen = scen\_id,  
 thresh = 0.1,   
 ylab = "Pr[min(SSB/SSB0) < 0.2] (Years 1 - 20)",   
 xlab = expression(gamma),  
 title = "Depletion",   
 scen\_label = "Scenario ID"  
 )  
  
# Plot for escapement level under considered gammas  
p\_esc <- plot\_gammas\_vs\_rule(  
 dt = openmse\_scen\_metrics,   
 gamma = gamma,   
 rule\_value = ESC,   
 scen = scen\_id,   
 thresh = 0.75,  
 title = "Escapement",   
 ylab = "med(SSB)/med(SSB0) in Final Year",  
 xlab = expression(gamma),  
 scen\_label = "Scenario ID"  
 )  
  
p\_dpl/p\_esc + plot\_layout(guides = 'collect')

|  |
| --- |
| Figure 3.3: Depletion probabilities and escapement levels at considered harvest rates under each scenario. Horizontal dashed lines indicate the 10% probability limit of the depletion rule (top plot) and the 75% critical threshold of the escapement rule (bottom plot). |

### 3.5.3 Estimated under each input scenario

**?@tbl-openmse-selected-gamma** shows the precautionary harvest rate estimated under the openMSE framework for each input scenario based on the 3-stage decision rule ([Section 2.1](#sec-grym-intro)).

Similarly to the results obtained under the Grym approach, the magnitude of estimates across the considered input scenarios is strongly influenced by the chosen Proportional Recruitment scenario. Scenarios using PR estimates reflecting lower variability and uncertainty in yearly recruitment () and (scn-1 & scn-2 under PR-emm21) produced the highest estimates , based on the escapement rule (). In contrast, scn-3, scn-4, scn-7 and scn-8, using PR estimates conveying the highest variability in (PR-amlr and PR-amlr-haul), returned estimates of 0 based on the depletion rule (). Therefore, under these scenarios, current management objectives would not be satisfied even in the absence of fishing.

The effect of maturity-at-length on estimates is also evident, with scenarios employing the mat-2021 ogive yielding lower estimates of compared to scenarios using the mat-2010 ogive. Under the mat-2021 ogive, the relatively higher proportion of immatures being vulnerable to fishing accelerates the decline of from , returning more conservative estimates.

openmse\_scen\_gamma\_select <- read\_rds("../part1\_shared\_files/outputs/openmse/openmse\_scen\_gamma\_select.rds")  
  
# gammas in math format  
gamma\_symb <- c("\\gamma\_1", "\\gamma\_2", "\\gamma\_p")  
  
openmse\_scen\_gamma\_select |>  
 select(-contains("approx")) |>  
 mutate(across(-scen\_id, as.numeric)) |>  
 mutate(across(c(gamma1, gamma2), ~if\_else(is.infinite(.), 0, .))) |>  
 left\_join(scenarios\_key, by = "scen\_id") |>  
 relocate(gamma1:selected\_gamma, .after = last\_col()) |>  
 mutate(gamma\_p = pmin(gamma1, gamma2)) |>  
 select(-selected\_gamma) |>  
 flextable() |>  
 set\_table\_properties(width = 0.8, layout = "autofit") |>  
 #bold(j = ~ gamma\_p) |>  
 bold(j = ~ gamma1, i = ~ gamma1 < gamma2) |>  
 bold(j = ~ gamma2, i = ~ gamma2 < gamma1) |>  
 vline(j = ~ gamma2, border = fp\_border\_default(width = .5)) |>  
 bg(j = ~ gamma\_p, bg = "#E1F5FE", part = "all") |>  
 set\_header\_labels(  
 scenario\_id = "Scenario ID",  
 pr\_scen\_id = "PR Scenario ID",  
 mat\_scen\_id = "Maturity ID"  
 ) |>  
 width(j = ~ gamma1 + gamma2 + gamma\_p, width = 10) |>  
 compose(j = ~ gamma1 + gamma2 + gamma\_p,   
 part = "header",   
 value = as\_paragraph(as\_equation(gamma\_symb))) |>  
 align(j = ~ gamma1 + gamma2 + gamma\_p, align = "center")

**Table** **:** Estimated precautionary harvest rate under each inputs scenario

| **scen\_id** | **PR Scenario ID** | **Maturity ID** |  |  |  |
| --- | --- | --- | --- | --- | --- |
| scn-1 | PR-emm21 | mat-2010 | 0.095 | **0.0875** | 0.0875 |
| scn-2 | PR-emm21 | mat-2021 | 0.070 | **0.0675** | 0.0675 |
| scn-3 | PR-amlr | mat-2010 | **0.000** | 0.0425 | 0.0000 |
| scn-4 | PR-amlr | mat-2021 | **0.000** | 0.0275 | 0.0000 |
| scn-5 | PR-atlantida | mat-2010 | **0.065** | 0.0700 | 0.0650 |
| scn-6 | PR-atlantida | mat-2021 | **0.050** | 0.0600 | 0.0500 |
| scn-7 | PR-amlr-haul | mat-2010 | **0.000** | 0.0250 | 0.0000 |
| scn-8 | PR-amlr-haul | mat-2021 | **0.000** | 0.0150 | 0.0000 |

### 3.5.4 Comparison between Grym and openMSE results

Projections of spawning stock status () simulated under the Grym implementation (obtained in [Chapter 2](#sec-grym-sims)) and the openMSE approach, for the considered input scenarios, are compared in [Figure 3.4](#fig-openmse-vs-grym-sss-scen) and [Figure 3.5](#fig-openmse-vs-grym-diff-med_sss). Key findings include:

* OpenMSE simulations demonstrate higher variability (i.e. greater uncertainty) in the annual estimates compared to the Grym approach. This is evident from the wider 90% quantile intervals observed in the openMSE results.
* The difference in dispersion between the two frameworks is more pronounced in scenarios scn-3, scn-4, scn-7 & scn-8, which are characterized by PR estimates with higher variability in .
* Moreover, for scenarios scn-3, scn-4, scn-7 & scn-8, dispersion levels estimated by the Grym implementation tend to decrease over the projection period at higher values, while dispersion remains stable in the openMSE simulations. This discrepancy may be attributed to the recruitment depletion penalization applied in Grym, which is not incorporated in the openMSE implementation (see [Section 3.2.2](#sec-openmse-diffs-sr)).
* Similarity between the two frameworks in terms of median annual is primarily determined by the chosen PR scenario. For instance, in scenarios scn-1 and scn-2 (under PR-emm21), the two implementations return matching trajectories of median up to harvest rates of , while for scn-5 and scn-6 (under PR-atlantida) the proximity in trajectories extends to harvest rates up to . However, in the remaining scenarios (under PR-amlr and PR-amlr-haul), the separation between the two frameworks occurs at lower harvest rates.
* The impact of the recruitment depletion penalization is also evident in the median trajectories in scenarios scn-3, scn-4, scn-7 & scn-8, where the decay in the projected median is faster under the Grym approach.
* The degree of impact of alternative maturity ogives on median projections appears to vary between frameworks as well. The differences in trajectories between scenarios using contrasting maturity ogives (e.g. scn-1 & scn-2, scn-3 & scn-4, etc) are more pronounced under the openMSE approach. This discrepancy is likely attributed to the difference in the computation of between the two frameworks. Grym calculates at a specific fraction of the year (“Spawning season”: days 76 to 138; **?@tbl-fixed-pars**) during which the stock biomass peaks due to peak individual growth. As a result, Grym tends to produce higher estimates of compared to the openMSE implementation, which estimates at the start of the year.

grym\_scen\_SSS <- grym\_scen\_outputs\_sub |>  
 rename(scen\_id = scenario\_id, sim = Run, gamma = Gamma) |>  
 select(scen\_id, sim, gamma, Year, SSS) |>  
 mutate(Framework = "Grym") |>  
 filter(Year != 0)  
  
openmse\_scen\_SSS <- openmse\_scen\_ssb |>  
 filter(gamma %in% gamma\_subset) |>  
 select(scen\_id, sim, gamma, Year, SSS) |>  
 mutate(Framework = "openMSE")  
  
frmwk\_scen\_SSS <- bind\_rows(grym\_scen\_SSS, openmse\_scen\_SSS)   
  
# Plot projections of spawning stock status under each gamma, across input  
# scenarios  
p\_SSS\_proj\_frmwk <- frmwk\_scen\_SSS |>  
 #filter(scen\_id == "scn-1", gamma == 0) |>  
 group\_by(scen\_id, Year, gamma, Framework) |>  
 median\_qi(SSS, .width = 0.9) |>  
 ggplot(aes(x = Year, y = SSS, fill = Framework, color = Framework)) +  
 geom\_lineribbon(aes(ymin = .lower, ymax = .upper), alpha = 1/3, linewidth = 0.3) +  
 geom\_line(linewidth = 0.3) +  
 scale\_fill\_manual(values = frmwk\_colours) +  
 scale\_colour\_manual(values = frmwk\_colours) +  
 facet\_grid(scen\_id ~ gamma, scales = "free") +  
 labs(y = expression(paste("Spawning stock status (SSB/", SSB[0], ")"))) +  
 theme(legend.position="bottom")  
  
ggsave(  
 plot = p\_SSS\_proj\_frmwk,   
 filename = "../part1\_shared\_files/outputs/plot\_openMSE\_vs\_Grym\_SSS\_traject\_scens.png",   
 width = 8,   
 height = 11  
)  
  
p\_med\_SSS\_proj\_frmwk <- frmwk\_scen\_SSS |>  
 #filter(scen\_id == "scn-1", gamma == 0) |>  
 group\_by(scen\_id, Year, gamma, Framework) |>  
 summarise(med\_SSS = median(SSS), .groups = "keep") |>  
 ggplot(aes(x = Year, y = med\_SSS, colour = Framework)) +  
 geom\_line() +  
 scale\_colour\_manual(values = frmwk\_colours) +  
 facet\_grid(scen\_id ~ gamma, scales = "fixed") +  
 labs(y = expression(paste("Spawning stock status (SSB/", SSB[0], ")"))) +  
 theme(legend.position="bottom")  
  
ggsave(  
 plot = p\_med\_SSS\_proj\_frmwk,   
 filename = "../part1\_shared\_files/outputs/plot\_openMSE\_vs\_Grym\_med\_SSS\_traject\_scens.png",   
 width = 8,   
 height = 8  
)

|  |
| --- |
| Figure 3.4: Median (solid lines) and 90% intervals (shaded ribbons) of simulated projections of Spawning Stock Status under the openMSE and Grym frameworks, for considered input scenarios (rows) and a subset of values. |

|  |
| --- |
| Figure 3.5: Median of simulated annual Spawning Stock Status in the projection period under the openMSE and Grym frameworks, for the considered input scenarios (rows) and a subset of values. |

The estimated harvest rates satisfying the depletion () and escapement () decision rules are compared between the two frameworks in [Figure 3.6](#fig-openmse-vs-grym-gammas).

In general, the openMSE implementation tends to produce estimates of and that are either slightly lower (scn-1, scn-2, scn-5 and scn-6) or identical (scn-4 and scn-8) to those estimated by the Grym implementation. For scenarios scn-3 and scn-7, the estimates of obtained in openMSE are marginally higher than those obtained in Grym. This reflects the faster decline in median trajectories under Grym, as observed and discussed in [Figure 3.5](#fig-openmse-vs-grym-diff-med_sss).

There is no clear pattern regarding the impact of alternative maturity ogives on and estimates from the two frameworks.

openmse\_gammas\_est <- openmse\_scen\_gamma\_select |>  
 mutate(Framework = "openMSE")  
  
grym\_gammas\_est <- grym\_gammas\_tbl |>  
 mutate(Framework = "Grym")  
  
frmwk\_gammas\_est <- bind\_rows(openmse\_gammas\_est, grym\_gammas\_est)  
  
gamma\_support\_tbl <- expand\_grid(  
 Framework = c("openMSE", "Grym"),  
 gamma = grym\_setups$gamma[[1]]  
)  
  
frmwk\_gammas\_est |>  
 pivot\_longer(cols = c(gamma1, gamma2), names\_to = "gamma\_type") |>  
 ggplot(aes(x = Framework, y = value)) +  
 #geom\_point(data = gamma\_support\_tbl, aes(x = Framework, y = gamma), size = 0.5, color = "gray80") +  
 geom\_point(data = gamma\_support\_tbl, aes(x = Framework, y = gamma), size = 1, color = "gray50", shape = "|") +  
 geom\_point(aes(color = gamma\_type, group = gamma\_type), size = 2) +  
 geom\_line(aes(color = gamma\_type, group = gamma\_type)) +  
 labs(  
 y = expression(paste("Harvest rate (", gamma, ")")),  
 x = NULL,  
 color = expression(paste("Type of ", gamma))  
 ) +  
 scale\_colour\_manual(values = c("#AFB42B", "#FF8F00"), name = "Harvest Rule",  
 labels = expression(gamma[1], gamma[2])) +  
 facet\_wrap(~scen\_id, ncol = 2) +  
 theme(legend.position="bottom") +  
 coord\_flip()

|  |
| --- |
| Figure 3.6: openMSE an Grym estimates of harvest rates complying with the depletion rule () and the escapement rule () for each of the considered input scenarios. Vertical gray ticks represent the values of considered for the analysis. |

Therefore, and considering the estimates of precautionary harvest rates presented in **?@tbl-openmse-vs-grym-selected-gamma**, results indicate a reasonable level of agreement between the two approaches across the considered input scenarios. The implemented openMSE-Grym approximation is capable of producing estimates of that are either identical or within a range of 1.25 - 2.25 percentage points lower than those obtained under the Grym’s base-case implementation.

gamma\_symb <- c("\\gamma\_p^{openMSE}", "\\gamma\_p^{Grym}")  
  
frmwk\_gammas\_est |>  
 group\_by(scen\_id, Framework) |>  
 mutate(gamma\_p = pmin(gamma1, gamma2)) |>  
 pivot\_wider(id\_cols = scen\_id,names\_from = Framework, values\_from = gamma\_p) |>  
 left\_join(scenarios\_key, by = "scen\_id") |>  
 relocate(pr\_scen\_id:mat\_scen\_id, .after = scen\_id) |>  
 flextable() |>  
 set\_table\_properties(width = 0.75, layout = "autofit") |>  
 set\_header\_labels(  
 scenario\_id = "Scenario ID",  
 pr\_scen\_id = "PR Scenario ID",  
 mat\_scen\_id = "Maturity ID"  
 ) |>  
 compose(j = ~ openMSE + Grym,   
 part = "header",   
 value = as\_paragraph(as\_equation(gamma\_symb))) |>  
 align(j = ~ openMSE + Grym, align = "center")

**Table** **:** Estimated precautionary harvest rates under the openMSE and Grym frameworks for the considered input scenarios.

| **scen\_id** | **PR Scenario ID** | **Maturity ID** |  |  |
| --- | --- | --- | --- | --- |
| scn-1 | PR-emm21 | mat-2010 | 0.0875 | 0.1075 |
| scn-2 | PR-emm21 | mat-2021 | 0.0675 | 0.0900 |
| scn-3 | PR-amlr | mat-2010 | 0.0000 | 0.0000 |
| scn-4 | PR-amlr | mat-2021 | 0.0000 | 0.0000 |
| scn-5 | PR-atlantida | mat-2010 | 0.0650 | 0.0775 |
| scn-6 | PR-atlantida | mat-2021 | 0.0500 | 0.0675 |
| scn-7 | PR-amlr-haul | mat-2010 | 0.0000 | 0.0000 |
| scn-8 | PR-amlr-haul | mat-2021 | 0.0000 | 0.0000 |

## 3.6 Discussion

The main objective of this analysis was to replicate the estimation of precautionary harvest rates for the Antarctic Krill, as currently performed under the Grym framework (Maschette et al., 2021; Maschette et al., 2020), using the R package {openMSE} (Hordyk et al., 2021). In order to achieve this, the components of openMSE (i.e. Operating Model, Management Procedures and Performance Metrics) were configured to approximate the modelling features of the Grym approach, as specified in CCAMLR’s [base-case](https://github.com/ccamlr/Grym_Base_Case/tree/Simulations) implementation for the Krill fishery. The effectiveness of the approximation was assessed by applying the openMSE implementation across a set of eight model input scenarios. These scenarios were designed to capture variations in Proportional Recruitment estimates (i.e. natural mortality, , and recruitment variability, ), as well as alternative maturity ogive curves. Each of these scenarios were also used in Grym simulations in a concurrent analysis ([Chapter 2](#sec-grym-sims)). By comparing the results obtained from openMSE with those from Grym under the same set of scenarios, we can examine the suitability of the openMSE framework to the Krill fishery.

Results revealed that the implemented openMSE approximation was able to generate estimates of precautionary harvest rates that were either identical or reasonably close to those obtained under the Grym implementation, for the considered input scenarios. Discrepancies in the estimates, as well as in the inherent estimates of harvest rates satisfying the depletion () and the escapement () rules, suggest that the openMSE approximation tends to yield slightly more conservative harvest rates than the Grym approach.

Discrepancies observed in results are likely due to fundamental structural differences between the two frameworks (see [Section 3.2](#sec-openmse-inherent-diffs)) that could not be addressed in the present analysis. For example, the absence of the recruitment depletion factor, the additional 2-year historical period under stochasticity, and the difference in the estimation of are all contributing factors for higher variability in openMSE simulations, increasing the chances of stock depletion falling below the critical threshold level and consequently leading to lower estimates of . On the other hand, openMSE is unable to approximate fine-scale temporal dynamics required for the calculation of key stock parameters at specific fractions of the year. For instance, Grym estimates specifically for the spawning season which, given the growth period under assumption, results in higher estimates compared to those calculated at the start of the year, as is the case with an annual model like openMSE. Consequently, Grym projections tend to return higher levels of escapement for the same level of exploitation, leading to higher estimates of .

The openMSE framework is still under active development, and it is possible that some of the missing Grym features could be incorporated with relative ease. However, increasing the temporal resolution to sub-year levels poses a significant technical challenge that would require substantial effort and development to implement.

The key strength of the openMSE framework is its capability to provide a comprehensive infrastructure to explore, evaluate and compare the performance of alternative management strategies. Furthermore, through its core package {DLMtool} (Carruthers and Hordyk, 2018), openMSE offers a range of Management Strategies tailored for data-limited populations, further enhancing its applicability to the Antarctic Krill fishery.

This analysis demonstrates that the implemented openMSE approximation is able to produce results that are fairly aligned with those obtained under the modelling framework currently adopted for the management of the Krill fishery. These findings instill confidence in the validity and reliability of the openMSE framework for conducting Management Strategy Evaluation (MSE) analysis for the Antarctic Krill.

Which is what we focus on next!

## 3.7 Supplementary Code

### 3.7.1 Operating Model builder function

# 4. Appendix: OM Report (*scn-1*)

An Operating Model for Krill under the openMSE approach

# References

Carruthers, T.R., Hordyk, A.R., 2018. The Data-Limited Methods Toolkit (DLMtool): An R package for informing management of data-limited populations. Methods in Ecology and Evolution 9, 2388–2395. <https://doi.org/10.1111/2041-210X.13081>

CCAMLR, 2022. [CCAMLR’s approach to managing the krill fishery. Euphausia superba in Area 48.](https://fishdocs.ccamlr.org/SAreport_48_KRI_2021.pdf)

Constable, A., de la Mare, W., 1996. A Generalised Model for Evaluating Yield and the Long-Term Status of Fish Stocks Under Conditions of Uncertainty. CCAMLR Science 3, 31–54.

Constable, A., de la Mare, W., Agnew, D., Everson, I., Miller, D., 2000. Managing fisheries to conserve the Antarctic marine ecosystem: practical implementation of the Convention on the Conservation of Antarctic Marine Living Resources (CCAMLR). ICES Journal of Marine Science 57, 778–791. <https://doi.org/10.1006/jmsc.2000.0725>

de la Mare, W., 1994. Modelling krill recruitment (No. CCAMLR Science 1).

Hordyk, A., Huynh, Q., Carruthers, T., 2021. [openMSE: Easily Install and Load the “openMSE” Packages](https://CRAN.R-project.org/package=openMSE).

Kawaguchi, S., 2016. Reproduction and Larval Development in Antarctic Krill (Euphausia superba), in: Siegel, V. (Ed.), Biology and Ecology of Antarctic Krill, Advances in Polar Ecology. Springer International Publishing, Cham, pp. 225–246. <https://doi.org/10.1007/978-3-319-29279-3_6>

Kinzey, D., 2021. Summary of proportional recruitment and multiyear biomass variability for krill in Subarea 48 from research surveys (No. WG-SAM-2021/21 Rev1). CCAMLR.

Maschette, D., Wotherspoon, S., 2021. Use of parameters within Euphausia superba Grym simulations (No. WG-FSA-2021/40). CCAMLR.

Maschette, D., Wotherspoon, S., Kawaguchi, S., Ziegler, P., 2021. [Grym assessment for Subarea 48.1 Euphausia superba populations](https://meetings.ccamlr.org/en/wg-fsa-2021/39) (No. WG-FSA-2021/39). CCAMLR, Hobart, Autralia.

Maschette, D., Wotherspoon, S., Pavez, C., Ziegler, P., Thanassekos, S., Reid, K., Kawagushi, D., Welsford, D., Constable, A., 2020. Generalised R Yield Model (Grym) (Scientific Committee No. SC-CAMLR-39/BG/19). CCAMLR.

Pavez, C., Wotherspoon, S., Maschette, D., Reid, K., Swadling, K., 2021. Recruitment modelling for Euphausia superba stock assessments considering the recurrence of years with low recruitment (No. WG-SAM-2021/09). CCAMLR, Hobart, Autralia.

SC-CAMLR, 2000. Report of the Bo Workshop (No. SC-CAMLR-XIX), Annex 4, Appendix G. CCAMLR.

Thanassekos, S., Reid, K., Kawaguchi, S., Wotherspoon, S., Maschette, D., Ziegler, P., Welsford, D., Watters, G., Kinzey, D., Darby, C., Trathan, P., Hill, S., Earl, T., Kasatkina, S., Ying, Y., 2021. Grym parameter values for Subareas 48.1, 48.2 and 48.3. (No. WG-SAM-2021/12). CCAMLR.

1. In addition to the configuring steps described in this section, we also [modified](https://github.com/Blue-Matter/MSEtool/compare/master...bcaneco:MSEtool:master) one of the internal functions of openMSE’s core package {MSEtool} to allow a minimum historical period of 2 years. To make the modified source code available for the simulations, {MSEtool} needs to be re-installed from the forked branch, e.g. via renv::install("https://github.com/bcaneco/MSEtool"). [↑](#footnote-ref-50)
2. In this case, ‘median’ refers to the median of the distribution of estimates. [↑](#footnote-ref-52)
3. Calculated as the proportion of simulations where the minimum value of yearly over the projection period is less than the 0.2 threshold. [↑](#footnote-ref-69)
4. Calculated as the ratio between the medians of simulated and values. [↑](#footnote-ref-70)
5. In addition to the configuring steps described in this section, we also [modified](https://github.com/Blue-Matter/MSEtool/compare/master...bcaneco:MSEtool:master) one of the internal functions of openMSE’s core package {MSEtool} to allow a minimum historical period of 2 years. To make the modified source code available for the simulations, {MSEtool} needs to be re-installed from the forked branch, e.g. via renv::install("https://github.com/bcaneco/MSEtool"). [↑](#footnote-ref-91)