Rescaling groundfish projected catch to meet the GOA OY cap

Alberto Rovellini

2025-03-04

This approach takes a vector of projected catch for a set of stocks and applies an optimization-based rescaling to transform it into a vector such that, given an ecosystem cap on total harvest , the following conditions are satisfied for each stock :

1. If , then

The input needed for the rescaling is a set of weights . The approach laid out here assumes that high-value stocks will have higher and will undergo lower rescaling. To satisfy condition 3 above, we need for all stocks.

A Shiny App illustrating this approach is available [here](https://swzlz7-alberto-rovellini.shinyapps.io/oy_rescaling_app/).

## Formulation

Our goal is to find a single multiplier that, when applied with the weights , produces rescaled ABCs that sum exactly to the cap. The approach first checks if rescaling is needed (i.e., if ). If not, no rescaling is performed and for all stocks.

If rescaling is needed, we calculate the reduction ratio :

This ratio tells us how much the total catch exceeds the cap.

We then search for a multiplier that, when used in the following rescaling equation, produces rescaled ABCs that sum to the cap:

Stocks with higher weights undergo less reduction because the exponent becomes smaller as increases. This preserves catch for high-value stocks while reducing catch more substantially for low-value stocks.

## Finding the multiplier

To find the optimal multiplier , we formulate an objective function that measures how close the sum of rescaled ABCs is to the cap:

The goal is to find the value of that minimizes this. We locate the optimal multiplier by means of numerical optimization using the bisection method (this assumes monotonic ). This is probably simplistic but it’s also something that we can easily code into Atlantis without more complex optimization approaches (I’d like to avoid having to call additional C libraries as that may break Beth’s code, also it seems like we should not need more complex optimization).

### The Bisection Method

The bisection method works by repeatedly dividing an interval in half and determining which half contains the minimum. The steps are:

1. Start with an interval where the minimum of is believed to exist (here arbitrarily using but should / could explore a different range)
2. Calculate the midpoint
3. Estimate the derivative of at using the central difference approximation:

* where is a small step size

1. If (within a tolerance), we’ve found the minimum
2. If , the minimum is in , so set
3. If , the minimum is in , so set
4. Repeat steps 2-6 until the interval is sufficiently small (tolerance set to 1e-7 for now, and 100 iterations as it should be a simple function)

This method converges to the optimal multiplier that balances the rescaling across stocks.