



Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Fisheries Research

journal homepage: www.elsevier.com/locate/fishres



Highlights

gadget3: An evolution of the gadget modelling framework

Fisheries Research xxx (xxxx) xxx

Jamie Lentin , Bjarki Elvarsson , William Butler , Valerio Bartolino 

- New fisheries assessment modelling framework, open-sourced on CRAN.
- Process-orientated structure ideally suited for MICE models.
- Configurable stock structure allows for length- and age-based stocks.
- Drivers can be added driven by external data sets.
- Entirely custom processes can be added to a model.

Graphical abstract and Research highlights will be displayed in online search result lists, the online contents list and the online article, but **will not appear in the article PDF file or print** unless it is mentioned in the journal specific style requirement. They are displayed in the proof pdf for review purpose only.

Full length article

gadget3: An evolution of the gadget modelling framework

Jamie Lentin^a , Bjarki Elvarsson^b , William Butler^b , Valerio Bartolino^c ^a Shuttle Thread Limited, Manchester, United Kingdom^b Marine and Freshwater Research Institute, Fornubúðum 5, 220 Hafnarfjörður, Reykjavík, Iceland^c Swedish University of Agricultural Sciences, Department of Aquatic Resources, Turistgatan 5, Lysekil, SE-453 30, Sweden

ARTICLE INFO

Handled by Andre Eric Punt

Keywords:

Fisheries management
Stock-assessment
Integrated analysis
Fish population dynamics
Simulation
Nonlinear models
Multispecies modelling

ABSTRACT

Fisheries systems vary widely in ecological processes, data availability, and management objectives, meaning that no single modelling structure is universally appropriate. Modellers frequently face limitations when adapting standardised frameworks to represent specific biological processes or to incorporate novel data types, while custom models tailored to individual systems can be difficult to reuse, maintain, or extend. Here we present gadget3, a new process-orientated modelling framework designed to address these challenges in modelling fish populations and fisheries dynamics. gadget3 is a complete re-implementation of the original GADGET framework, retaining its modular philosophy while extending and generalising its capabilities. gadget3 takes advantage of the R ecosystem of model optimisation utilities, particularly the Template Model Builder (TMB) library, enabling efficient parameter estimation through automatic differentiation and allowing advanced statistical features such as random effects. We outline the structure of a gadget3 model, which is assembled from modular components representing biological processes, fleet interactions, and likelihood components. Through the use of metaprogramming and composable functions, users can implement alternative formulations or entirely new processes while maintaining compatibility with the broader modelling structure. Examples are presented demonstrating how model behaviour can be adapted through custom formulations, allowing the incorporation of system-specific biological relationships, environmental drivers, or alternative recruitment and mortality functions. Finally, we discuss ongoing and planned developments, including expanded modelling capabilities, improved diagnostics, and enhanced support for complex multi-species and ecosystem applications. Together, these features position gadget3 as a flexible and extensible framework for developing next-generation fisheries and ecosystem models.

1. Introduction

Models of fish populations and fisheries dynamics are widely used to investigate the effects of management decisions, including area closures and catch allocations. These models have increasingly been developed using generic software packages that provide the features needed for an integrated analysis of available data (Punt et al., 2020), such as Stock Synthesis (Methot and Wetzel, 2013) and SAM (Nielsen and Berg, 2014). The capability of the software packages have been further advanced by the development of generic open source libraries such as ADMB (Fournier et al., 2012), TMB (Kristensen et al., 2016) and STAN (Carpenter et al., 2017) that provide generic approaches to parameter estimation and inference. This allows a number of downstream models to reap the methodological benefits made available by these generic packages.

The availability of standard assessment packages have greatly facilitated the uptake of statistical assessment methods in the provisioning

of advice, even in data-limited situations as noted by the recent uptake of SPiCT (ICES, 2024, and similar processes) and Gadget for various stocks where traditional age-based data is limited (Þór Elvarsson et al., 2018; ICES, 2023). In addition, these developments answer the need to fulfil the goal of sustainable fishing practices around the world (Fisheries, 1995; UN, 2002) by providing a basis to give advice within a precautionary framework such as developed by ICES (2019).

Fisheries models are however not a “one size fits all” because of the diversity in species ecology and adaptations that influence their population dynamics and exposure to fishing. Moreover, availability and properties of data collected on particular fish stocks can be largely different across species and regions. Modellers often have limited possibility to adapt models to the characteristics of the study system and need to shoehorn the data to fit the features of the software package or simply exclude relevant data. Surplus production models

* Corresponding author.

E-mail addresses: lentinj@shuttlethread.com (J. Lentin), bjarki.elvarsson@hafogvatn.is (B. Elvarsson), will.butler@hafogvatn.is (W. Butler), valerio.bartolino@slu.se (V. Bartolino).<https://doi.org/10.1016/j.fishres.2026.107903>

Received 27 April 2026; Received in revised form 14 October 2026; Accepted 14 October 2026

Table 1
Published advice utilising gadget3.

Stock	ICES advice code
Icelandic slope beaked redfish	reb.27.5a14 (ICES, 2023)
Blue ling in Icelandic and east Greenlandic waters	bli.27.5a14 (ICES, 2025)
Greenland halibut in Icelandic, Faroese, and east Greenlandic waters	ghl.27.561214 (ICES, 2023)
Greenland halibut in the NE Arctic	ghl.27.1-2 (ICES, 2023)

(e.g., Pedersen and Berg, 2017) rely solely on catch and indices of biomass, age based models (Nielsen and Berg, 2014; Miller and Legault, 2015) often require strong assumptions on availability of age data to calculate catch at age via age-length keys (Friðriksson, 1934) or using methods such as those described in Ailloud and Hoenig (2019) whenever data is missing on age.

Bespoke models such as described by Gudmundsson (1994), one of the first models where random deviations from the catch equation were considered, are likely developed with a specific application in mind. Taking these models and then applying to other scenarios can reveal major constraints, and their use often depends on access to the model developers or experienced users. While bespoke models may be precisely tailored to a particular case, model development can be time consuming, the end result lacking generality and flexibility by design. In contrast, standard software packages provide familiarity among modellers, along with standardised diagnostics and consistent data input and output formats (e.g., Taylor et al., 2021). A standard software package is also likely to have a more thorough test suite to ensure correct implementation and plans for long-term maintenance and extensibility (Punt et al., 2020).

Here, we present the development and design of gadget3, a flexible modelling framework for fish populations and fisheries dynamics. The framework is designed as a generic tool for modellers to better capture fish—fisheries system and data characteristics by implementing the required processes and interactions from components whilst allowing the flexibility for the user to add missing features if not already available. Integrating custom processes into a standard framework allows for the creation of standardised diagnostic tools and ensures familiarity amongst model developers whilst retaining flexibility of bespoke models. gadget3 is a complete re-implementation of GADGET (Begley and Howell, 2004), while incorporating recent developments from open source libraries such as TMB (Kristensen et al., 2016) and R (Team, 2025).

gadget3 and its ancillary packages are all available on GitHub under the “gadget-framework” organisation:

- **gadget3**: Core package, with stable releases also published to CRAN.
- **gadgetutils**: Utilities related to model optimisation (Butler et al., 2025)
- **gadgetplots**: Generates standard plots of gadget3 output (Vih-takari et al., 2025)
- **gadget2to3**: Translates gadget2 model files to gadget3 (Lentin and Elvarsson, 2022)
- **g3experiments**: Experimental/beta process implementations, a place to try ideas

gadget3 is already being used to produce assessment models for advice. Table 1 contains references to advice sheets backed by a gadget3 model.

2. History of gadget

The gadget3 R package is the culmination of several related projects under the Gadget framework banner, see Fig. 1. The original GADGET (Begley and Howell, 2004), retroactively renamed gadget2, is a fork of

Bormicon (Stefánsson et al., 1997) combining improvements made in Fleksibest (Guldbrandsen Frøysa et al., 2002).

The defining feature of gadget2 was its process-orientated, modular design, in which sections of the model could be added and removed to construct an integrated model of smaller components. This design enabled a wide range of model configurations within the same tool. Many sections supported switchable methods to change behaviour, which could be customised further with a minimal programming language to encode custom formulas. Aggregation files allowed multiple data sources to be incorporated as observation data within the same model. Finally, a detailed and well-written user guide described each process mathematically (Begley, 2005).

The MFDB R package (Lentin and Elvarsson, 2021), and its predecessor DSTDW (Taylor, 2005), are data warehousing tools. Using databases such as PostgreSQL, these packages are used to store observation data and to subsequently produce inputs for ecosystem modelling tools via data aggregation, transformation, and sampling functions. The primary aim of both is to manipulate and transform data required by gadget2 configuration files, and generate required data files. The database is not intended as a permanent store of information, but a staging area for data that can potentially be used within the model. This approach allows source data to be sculpted and re-sculpted to suit new model formulations programmatically, improving model development speed.

The paradigm of configuring gadget2 from R spawned the Rgadget package (Elvarsson and Lentin, 2020), which integrates gadget2 configuration, execution, and reporting within an R environment. A detailed understanding of the gadget2 configuration format was no longer necessary, and the creation of observation data could be interleaved with model construction and provided directly as data frames. Repetitive structures could be wrapped in functions, making adaptation of an existing model to a new scenario a much simpler prospect.

3. gadget3 inception & design decisions

The primary aim of the gadget rewrite was to take advantage of the R (Team, 2025) ecosystem of model optimisation utilities that has since emerged, in particular TMB (Kristensen et al., 2016), which provides a simple framework for optimising models using automatic differentiation (AD). AD promises substantial benefits in convergence time for models, one of the main criticisms of gadget2. Adapting the gadget2 codebase, removing conditionals for compatibility with AD, would have not been possible without a rewrite. Besides AD, TMB offers many facilities useful in a fisheries modelling context, i.e., random effects, multivariate distributions, one step ahead residuals, amongst others.

Whilst gadget2 and gadget3 do not share any code, all gadget2 concepts have a one-to-one mapping to gadget3. The aim has been to preserve the strengths of gadget2 in gadget3; the modular structure that defined gadget2 is still present and has been further generalised for rapid addition of new concepts.

The other key aim was to increase flexibility, and loosen restrictions inherent in the gadget2 structure where possible. For instance, in gadget2 there was a single natural mortality process for the entirety of the stock. Separate drivers for natural mortality would have to be overlaid

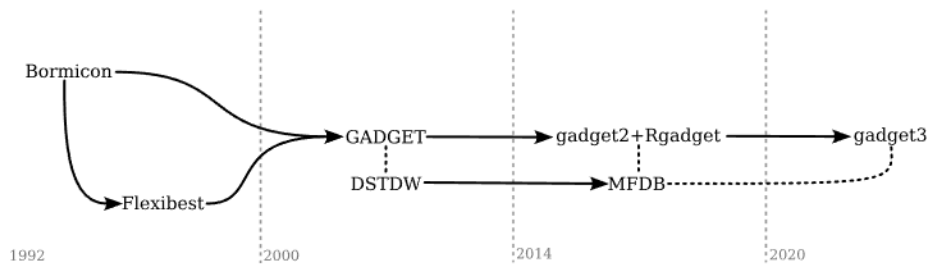


Fig. 1. Schematic of the gadget family tree. Solid lines indicate inheritance of code/ideas into a new tool, dotted lines demonstrate compatible supporting tools. Years on the time line correspond to major developments.

on top of each other to be included in the model. *gadget3* allows for any number of natural mortality processes, targeted at different subpopulations, allowing mortality drivers to be modelled separately.

Within a timestep, the order in which processes were applied in a *gadget2* model are fixed. Generally this does not pose an issue, however this can be limiting for advanced users. For instance, one may want to ensure that natural mortality events happen before or after spawning, or use reporting to inspect abundance mid-timestep. *Gadget3* defaults to an ordering matching *gadget2*, however this ordering can be customised if required.

In *gadget3* a model is assembled by treating code as data, an approach known as metaprogramming, in effect treating everything as a gadget formulas. Custom functions within a process can be implemented without any modifications to core code, or having to re-implement the scaffolding required for a mortality process. The details of this will be explored in Section 5.

The abstract code of a *gadget3* model is generally assembled into a C++/TMB function for optimisation. However, the code can also be turned into a native R function, and potentially other formats in future. These function outputs are designed to be human-readable, stand-alone functions that can be inspected in the context of the current model without having to work through the underlying *gadget3* implementation layers and abstractions.

Because function output is generated specifically for a given scenario, the resulting model code contains very little branching logic (e.g., if statements). Methods are not selected at run time, only the methods required for the scenario are included in the generated code. To apply automatic differentiation TMB represents any function as a chain of differentiable operations leading to the output, known as the “tape”. During optimisation, this sequence is assumed to remain fixed. Branching logic can violate this assumption because the sequence of operations may change depending on the input values. By minimising branching in the generated code, potential incompatibilities with automatic differentiation become easier to identify and avoid.

gadget3 defines models entirely within R. Model structure, customisations and data are all formulated using R code and can be produced programmatically without a helper library. Common setup (e.g., construction of multiple fleets) can be wrapped by functions instead of duplicating configurations and observations are provided as simple R data frames by integrating data extractions from MFDB as well as from any other source. A number of additional benefits are also achieved by integrating *gadget* within the R ecosystem: models are more easily absorbed into existing R-based assessment/analysis workflows, and established software development practices can be adopted, for instance standardised package distribution via CRAN. Moreover, when writing *gadget3* models, users can take advantage of integrated development environments (IDE) providing auto-completion and context-dependent documentation for *gadget3* concepts instead of requiring knowledge of configuration file formats.

4. Structure of a gadget3 model

A *gadget3* model is defined by a list of *actions*, each action defining a process in the model or a likelihood component. The functions to generate these actions have the prefix *g3a_**, for instance, *g3a_spawn()* adds spawning, *g3a_predate()* adds predation, and *g3a_time()* is the general timekeeping for a model.

gadget3 actions operate on stock objects. In Fig. 2 we define separate stock objects for immature and mature ling, with different length groups so we can capture their growth patterns. As with *gadget2* (sub)stocks, a stock object does not necessarily represent an actual biological stock but any homogeneous group within a population. Stock objects are also commonly split between sexes or an asymmetric combination of maturity stage and sex for instance “all-male”, “immature-female”, “mature-female”.

Internally they define the dimensions of arrays used to store a stock’s state, in the example model in Section 6.1 immature ling will be divided into length groups from 20 to 112+ cm in 4 cm intervals, and ages will span 5 to 20. Additional dimensions can be stacked on top to capture relevant processes, for instance, *g3s_livesonareas()* assigns an area or multiple areas to a stock, and *g3s_tag()* adds a tag dimension to record tagging experiments.

A fleet, defined with *g3_fleet()*, is just a stock object without length structure. As a result, *g3s_livesonareas()* can be applied to it in the same way when creating a multi-area model.

Fig. 2 defines a single action using *g3a_predate()*, the stock objects are provided for both the predator and its preys, in this case the action defines the harvesting of both stocks by the commercial fleet. Instead of customising the fleet behaviour by choosing from a list of available options, in *gadget3* this customisation is done via *function composition*, chaining together simple functions to form more complex behaviour. In the above, *g3a_predate_catchability_total_fleet()* instructs our predator to extract a total biomass, and *g3_timeareadata()* provides a table of landings data to use as this value. The argument supplied to the *catchability_f* parameter could be swapped out with other available or customised functions without having to alter the main predation code, for instance, supplying *g3a_predate_catchability_linearfleet()* would create a linear harvesting strategy where the relationship between available biomass and catches is determined by a scaling factor.

A stock object can also abstract interpolating its own structure to another stock. This is used when transferring maturing individuals from an immature to mature stock object. Consequently, a stock can use a highly granular length dimension where it is necessary to resolve biological details (e.g., the condition of a mature stock), while a coarser resolution can be used where such details are of less importance (e.g., an immature stock under low selective pressure).

These conversions are also used when translating model data for likelihood components, meaning stock structure does not have to match the format of landings data. Multiple model stocks can also be combined when comparing to observations, for example immature & mature stocks will be combined if landings data does not differentiate between the 2 groups. Stock and data resolution on all dimensions (length, age, area, fleet) are relatively independent and are reconciled internally for the purpose of fitting.

```

stocks <- list(
  imm = g3_stock(c(species = 'ling', 'imm'),
    lengthgroups = seq(20, 112, 4)) |>
    g3s_age(minage = 5, maxage = 20),
  mat = g3_stock(c(species = 'ling', 'mat'),
    lengthgroups = seq(36, 140, 4)) |>
    g3s_age(minage = 5, maxage = 20) )

fleets <- list(
  comm = g3_fleet('comm'),
  surv = g3_fleet('surv') )

actions <- list(
  g3a_predate(
    fleets$comm,
    stocks,
    suitability = g3_suitability_exponential150(
      by_stock = "species" ),
    catchability_f = g3a_predate_catchability_totalfleet(
      g3_timeareadata('comm_landings', expand.grid(
        year = 1990:2025,
        step = 2,
        total_weight = 5e5 )))))

```

Fig. 2. Defining stocks, fleets, and a predation interaction in gadget3. Two stocks (immature and mature ling stocks) and two fleets (commercial and survey fleets) are defined; the predation action captures the harvesting of both stocks by the commercial fleet.

4.1. Available actions: Stock dynamics

gadget3 comes with a suite of actions for implementing stock dynamics, see [Table A.1](#) for the full list currently available. In the majority of cases, these are re-implementations of existing gadget2 methods to allow existing models to be quickly re-implemented, but there are some with no parallel in gadget2. For example, the `g3a_spmode()` action allows the dynamics of a stock to be modelled with a surplus-production model, as a half-way house between otherfood (fixed abundance) and fully modelled stocks. Regardless, the behaviour of these actions can be customised further than what was possible in gadget2. See [Section 5](#) for more information.

4.2. Available actions: likelihood

A set of gadget3 actions named `g3l_*` are responsible for adding likelihood components to a model, see [Table A.2](#) for the full list. All of these actions contribute to a cumulative negative log-likelihood (nll) variable that is returned when the model is evaluated.

Most of gadget2's likelihood components are now subsumed into `g3l_catchdistribution` and `g3l_abundancedistribution`. In gadget2, separate aggregation files were required to ensure model data was aggregated to match the structure of the corresponding observation data frames. In gadget3, aggregation is inferred directly from the columns and data present in supplied observation data frames. For example, age—length observations are differentiated from length-only data by the presence of an “age” column, while length bins are derived from the values in the length column itself. When the model is executed, model data is aggregated into a corresponding array, enabling direct comparison with the observation data.

There are also entirely new likelihood actions, notably `g3l_sparsesample_*`. In contrast to `g3l_catchdistribution` which assumes a large dataset has been aggregated into bins, `g3l_sparse sample_*` takes a set of individual observations, usually a biological metric or attribute, and compares them to the mean biology of the corresponding group of individuals in the model. Whilst the measurement could be standard (e.g., weight), there is scope for including a custom calculation. The motivating example here is fitting blubber thickness

observations for marine mammals within gadget3, using an empirical relationship between blubber thickness, length and weight ([Ryg et al., 1990](#)).

4.3. Parameterisation & execution

The list of actions is converted into a concrete function using either `g3_to_tmb()` or `g3_to_r()`. The former produces C++ code suitable for compilation with TMB, the latter produces a standalone R function that can be used for single model runs. In both cases, the output of the resultant functions will be identical. However:

- The R function has no associated gradient function, so is ill-suited for optimisation runs.
- The TMB code has to be compiled into a binary before execution, a process that can be slow. The R function allows you to iterate faster when developing the model.
- As a regular R function, it can be debugged like any other R code, e.g., editing with `fix()`, inserting `print()` statements or using `browser()`.
- Regular R code can be integrated to the model when using the R function, allowing for feedback loops as required for management strategy evaluations.

Also produced is a *parameter template* data frame. This needs populating with initial values, bounds and optimisation settings for every parameter found in the model (the mechanism for adding parameters to the model is discussed in [Section 5](#)). For the R version, this can be used directly as input to the function to perform a single model run. For the TMB version the completed parameter template is passed to `g3_tmb_adfun()`, which constructs a TMB objective function ready to optimise selected parameters.

The optimisation process can be run using base R's `optim()`, as any other TMB model. In the practice of implementing gadget models, the problem of data weighting is typically addressed with an iterative reweighting approach as introduced by [Stefansson \(2003\)](#) and subsequently implemented in [Taylor et al. \(2004\)](#) and [Elvarsson and Lentin \(2020\)](#). The function `g3_iterative()` from [gadgetutils](#) provides a

gadget3 version of this procedure embedding both data weighting and optimisation.

Assuming a successful optimisation, `g3_fit()` from `gadgetutils` generates convenient model summaries for diagnostics and inspection of results. Finally, the results of a model run can be compiled into an HTML report containing standard plots using `gadget_plots()` from the `gadgetplots` package.

5. Customising gadget3 models

We saw earlier that gadget3 actions are produced by composing functions. The output of `g3_timeareadata()` is fed into `g3a_predate_catchability_totalfleet()`, to `g3a_predate()`. Parameterisation was also added using the generic helper `g3_parameterized()`.

In each instance, the input and output of each of these functions are R formula objects. An R formula is a combination of unevaluated R code and an environment that can contain variable definitions. The unevaluated code is an *abstract syntax tree*, i.e., a nested structure with one entry per function call in the code. gadget3 functions manipulate these structures rather than running code directly, most commonly substituting the given formula into its own formula to return.

A function like `g3a_predate_catchability_totalfleet()` essentially places the code chosen for catchability in the relevant slots of its own code. A separate optimisation step then hoists any code that does not need to be recomputed for each age/area by inspecting free variables and moving it outside the relevant loops. The implementer of a growth function does not need to inform gadget3 if a growth function is dependent on age for instance, the fact that it mentions age is sufficient. As a result, the implementations of most gadget3 functions are very close to their mathematical representations.

gadget3 provides a set of interlocking functions, but these do not use any special functionality unavailable to the end user. Writing your own formulas, or customising existing ones with custom code, can be done using the `g3_formula()` low-level helper.

gadget3 makes common helper functions available to models. One such function is `avoid_zero()`, which uses log-space addition to create a continuous function that is linear only for positive values. This and similar functions, such as `dif_pmin()`, allow values to be bounded without using if statements.

Fig. 3 provides several examples of how gadget3 behaviour can be modified.

Firstly, we use `g3_formula()` to modify `g3a_natural_mortality()` to use two different values of M , before and after a specific point in time. Note that we have not had to re-implement any of the code that manages the removals from the model, only the code that produces a value of M .

In the next example, this is taken a step further and `g3a_natural_mortality()` is modified to implement a separate density-dependent pup survival function. Time-series data of ice cover is imported using `g3_timeareadata()`, informing a relationship to pup mortality. The formula uses a stock variable (`rin_mat_num`) creating a dynamic internal feedback mechanism as it is typical of density-dependent processes. Separately we define a formula for when & where this applies. Note that we can have multiple natural mortality actions, each implementing a different effect and source of natural mortality.

Other actions can be customised in a similar fashion. The next example provides a `g3_formula()` to `g3a_spawn()` to implement the general concept of body condition thresholds (Adamczak et al., 2023) which has gained large attention in the study of marine mammals (e.g., Pirota et al., 2018).

Fig. 4 demonstrates how the modified Beverton–Holt recruitment function from stock synthesis 3 (Methot and Wetzel, 2013) is implemented in gadget3.

As with most of the gadget3 standard library, the function returns a formula with its free parameters bound to the function arguments. If you do not supply any extra arguments the formula is parameterised in the model, but each could be customised further by supplying a formula.

5.1. Models of intermediate complexity for ecosystem assessments (MICE)

MICE models as defined in Plagányi et al. (2014), sit in-between a traditional single-species model and a whole-ecosystem model, e.g., Atlantis (Fulton et al., 2004). A MICE model should be able to capture critical interactions driving the dynamics of populations. These interactions can include both internal and external processes (for example density-dependency, predation or the effect of an abiotic driver), without invoking the complexity of a whole-ecosystem model.

Looking for a few critical interactions stimulates thinking of model structure in terms of processes and results in a more realistic model that can be more intuitively understood. Ultimately, it is a step towards process-oriented modelling. This was a niche that gadget2 was designed to fill, and has been used to simulate complex species interactions in various ecosystems such as the interactions between cod; herring and sprat in the Baltic Sea (Kulatska et al., 2019), between minke whale; cod; herring and capelin in the Barents Sea (Lindström et al., 2009), between cod; redfish and shrimp in the Flemish Cap (Pérez-Rodríguez et al., 2017).

As a gadget3 model is a list of actions, concatenating two or more lists of actions representing distinct species is technically all that is required to produce a multispecies MICE model. These separate lists of actions can be well-defined models in themselves, allowing development and optimisation of single-species models in isolation and then use both the actions and optimised parameters as starting points for the combined model. An important aspect of an estimation strategy for large models and for reducing the time taken to converge.

The combined model can then have further interactions added. Top-down effects can be introduced by adding predation between stocks. This can be achieved with `g3a_predate()` as demonstrated earlier but with `g3a_predate_catchability_predator()` instead of one of the fleet catchability functions. Bottom-up effects are also built-in to gadget3; the formula produced by `g3a_grow_weight_weight_jones()` depends on `stock_feedinglevel` to introduce a prey-density dependence on consumption rate and ultimately on growth. Internal interactions can be represented by adding a custom dependence on stock condition, e.g., `stock_num` or `stock_wgt`, as introduced earlier when adding a fertility-body condition relationship for seals or also via intra-specific food competition emerging by `g3a_grow_weight_weight_jones()`.

5.1.1. Alternative stock structures

Neither the predator nor the prey stock is required to have a full age–length structure in gadget3. There may not be enough data to inform the stock structure for the stock, for instance, or in keeping with MICE philosophy a full stock structure may be unnecessary for the question of interest.

Defining a stock without length and/or age structure is a case of omitting these parts from the definition. There are two options for modelling a stock without both age and length structures in gadget3:

- `g3a_otherfood()` for an externally-estimated stock without any structure, abundance amounts can be specified as parameters. Closely related to the otherfood concept in gadget2.
- `g3a_spmodel()` for a stock estimated internally with a simple surplus production model.

The `g3a_spmodel()` action is a useful addition for multi-species models because it provides the option to model dynamic prey with simple structures (no age or length dimensions), offering a middle ground that is more realistic biologically than otherfood prey items (which are non-dynamic) and simpler, and thus more computationally efficient, than a fully-structured stock.

```

# Parameterise with two different values of M, before & after a
# breakpoint, the latter being age-dependents
g3a_naturalmortality(stocks$imm, g3a_naturalmortality_exp(
  g3_formula(
    if (cur_year < 2020) m.early else m.late,
    m.early = g3_parameterized('M.early'),
    m.late = g3_parameterized('M.late', by_age = TRUE) )))

# Density-dependent seal pup survival function: Natural mortality
# based on ice cover, predictions from external model
ice_cover_timeseries <- expand.grid(year = 1990:2099, value = 1)
g3a_naturalmortality(stocks$imm,
  g3a_naturalmortality_exp(
    g3_formula(
      avoid_zero(p1 * (sum(rin_mat_num)/ice_cover - p2)),
      ice_cover = g3_timeareadata('ice_cover_obs',
        ice_cover_timeseries, 'value'),
      p1 = g3_parameterized('steepM.rin',
        value = 10, optimise = FALSE ),
      p2 = g3_parameterized('breedDens.rin',
        value = 1.18, optimise = FALSE )),
    run_f = g3_formula(
      (cur_step == 1 || cur_step == 2) && age == stock_minage ))

# Fertility in seals as a function of body condition
g3a_spawn(
  stocks$mat,
  output_stocks = list(stocks$imm),
  recruitment_f = g3a_spawn_recruitment_fecundity(),
  proportion_f = g3_formula(
    avoid_zero(1 - avoid_zero((1 + minSpwTh) -
      (1 + minSpwTh) * stock_ss(stock_wgt)/
      (walphaRef * stock_midlen^wbetaRef))),
    walphaRef = g3_parameterized('walphaRef.rin'),
    wbetaRef = g3_parameterized('wbetaRef.rin'),
    minSpwTh = g3_parameterized('minSpwTh.rin') ))

```

Fig. 3. Examples of implementing custom processes in gadget3.

```

g3a_spawn_recruitment_bevertonholt_ss3 <- function (
  # Steepness parameter
  h = g3_parameterized("spawn.h",
    lower = 0.1, upper = 1, value = 0.5, by_stock = by_stock),
  # Recruitment deviates
  R = g3_parameterized("spawn.R",
    by_year = TRUE, exponentiate = TRUE, scale = "spawn.R0",
    by_stock = by_stock),
  # Unfished equilibrium spawning biomass (corresponding to R0)
  B0 = g3_parameterized("spawn.B0", by_stock = by_stock),
  # Prepend the stock name to parameter names?
  by_stock = TRUE
) {
  list(
    s = ~sum(stock_ss(stock_wgt) * stock_ss(stock_spawningnum)),
    r = ~4 * h * s * R/(B0 * (1 - h) + s * (5 * h - 1))
  )
}

```

Fig. 4. Implementation of modified Beverton–Holt recruitment from SS3 within gadget3.

5.2. Projections / MSE

Whilst `gadget2` is ultimately deterministic, `gadget3` can produce stochastic simulations. The `g3_param_project()` function can be used to insert a time-series parameter to represent deviations, the distribution selected will (1) add a likelihood to fit values to the distribution and (2) supply new values from the distribution when projecting the model. The blue ling model (see Section 6.1) uses `g3_param_project()` to integrate out recruitment deviations and provide autocorrelated projections.

`g3a_predate_catchability_project()` can be used to incorporate projections into fleet landings, automatically switching between landings data for the historical period and a quota formulation for projections. Quotas are defined using `g3_quota()`, this handles the mechanics of defining the fishing calendar (when an assessment is produced, when the quota is applied, and the period of the entire process). The assessment function produces a Total Allowable Catch. This function can be a simple hockeystick curve based on current abundance or an external R function that applies a separate model.

This design allows `gadget3` to serve as a tool to implement different parts of a full Management Strategy Evaluation framework; it can equally provide an operating model and framework for evaluation of management procedures, or the assessment model as part of the management procedure.

6. Results

6.1. Example model: Blue ling

Blue ling (*Molva dypterygia*) in the waters around Iceland and East of Greenland are assessed as one stock. Blue ling is a deep-water gadoid species primarily distributed along the continental shelves and slopes of the Northeast Atlantic, particularly around Iceland, the Faroe Islands, Norway, the British Isles, and off East Greenland. In Icelandic waters, it is commonly found in deeper regions over soft seabeds in the southeast, south, and west of Iceland, at depths greater than 500 m. The majority of the population is concentrated between 300 and 800 m, and it is rarely found at depths shallower than 200 m. During the winter months, blue ling tends to stay at deeper depths, while in the spring and summer, it ascends to shallower areas.

Appendix B contains a full breakdown of the model structure of an assessment model for blue ling using `gadget3`, along with the output produced. The R code used to generate the model can be found in the supplementary data.

The model is fitted to several independent data sources. Length-disaggregated survey indices were derived from the Icelandic Autumn Groundfish Survey (IAGS), the Greenland halibut survey in east Greenland (GHLE), and the shrimp survey in east Greenland (SFE). The three indices were standardised using the Cochran method (Cochran, 1977) and combined into a single index covering Icelandic (27.5a) and east Greenlandic (27.14) waters. For use in the assessment, this combined index was disaggregated into seven length bins: 20–44, 44–56, 56–68, 68–80, 80–92, 92–104, and 104–140 cm. The remaining inputs comprised length distributions from the Icelandic commercial bottom trawl and longline fleets, quarterly landings from Iceland and Greenland, and maturity data from the autumn survey. An overview of the input data and their annual availability is shown in Fig. 5.

Also present is a model variant that treats annual recruitment deviations as random effects, integrated out by the Laplace approximation (via TMB). The log-AR(1) process Eq. (1) serves as the prior:

$$\log R_y \mid R_{y-1} \sim \mathcal{N}(\varphi \log R_{y-1} + (1 - \varphi) \log \bar{R}, \sigma_R^2) \quad (1)$$

Fig. 6 demonstrates the differing recruitment patterns between the model variants. The random effects model produces smoother, more autocorrelated recruitment trajectories than the fixed effects model, reflecting the regularising effect of the log-AR(1) prior; year-to-year

variation is partially absorbed by the estimated autocorrelation and standard deviation parameters. Despite these differences in recruitment, both models produce similar fits to the survey indices across all length strata (Fig. 7), indicating that the data do not strongly favour one formulation over the other on the basis of fit alone. The primary advantage of the random effects formulation is methodological: it integrates over recruitment uncertainty when estimating other fixed-effect parameters (selectivity, growth spread, maturation ogive), and provides a principled autocorrelation structure for forward projections.

6.2. Gadget2 vs gadget3: Greater silver smelt

Greater silver smelt is a small silvery bathypelagic species that can form large schools close to the seafloor mainly at depths >500 m. In Icelandic waters it can live to around 26 years old. Juveniles tend to aggregate in shallower depths. Greater silver smelt mainly feed on zooplankton (e.g., euphausiids, amphipods, and copepods) or small nekton (e.g., squids, jellyfish, or fish). `gadget2` model has been previously used as an assessment model, running from 1990 to 2023. Greater silver smelt is modelled as an immature and mature sub-stock with ages 1–14 and 3–26 respectively. Length is binned from 8 cm to 60 cm into 1 cm bins. Data is sourced from commercial catch samples and the Icelandic autumn groundfish survey.

As `gadget3` concepts closely match `gadget2` options, it is possible to mechanically translate from one to the other. The `gadget2to3` package can translate `gadget2` model configuration into an equivalent `gadget3` model. Using this against the existing greater silver smelt model configuration, we can optimise the same model using both `gadget2/rgadget` and `gadget3/gadgetutils`.

Tables 2 and 3 breakdown the iterative re-weighting process for both the original `gadget2` model and the translated `gadget3` model. In both cases optimisation is done using best practices for each (a combination of algorithms for `gadget2`, BFGS & 2 stages of reweighting for `gadget3`), rather than trying to use directly comparable optimisation methods. Within each stage, all optimisations run in parallel, thus the slowest process determines the total run time. The `gadget3` process finishes in under an hour, vs. ~2.5 h for `gadget2`. The speed improvement stems from `gadget3`'s use of gradient information via automatic differentiation: `gadget2` required approximately 400,000–750,000 function evaluations per component in stage 1 alone, whereas `gadget3` reached comparable results with 10,000–21,000 function evaluations — roughly two orders of magnitude fewer. The final NLL scores are broadly comparable across both models, consistent with the `gadget2to3` translation preserving model structure.

7. Software development processes

The intention with `gadget3` has always been to run the project using an open-source development model, and use software development best-practices.

The `gadget3` repository includes an extensive test suite. As `gadget3` models can be pared down to a minimal number of actions we can use this to create unit tests, isolating the behaviour of individual actions. Tests then ensure that action behaviour matches what is expected under various conditions. Utilities then compare output of the R and TMB implementations for each of these tests, ensuring identical behaviour between the two. As we have one-to-one compatibility with `gadget2`, we can also run integration tests that run a `gadget2` model, automatically create an equivalent `gadget3` model and compare the output.

Whilst available on GitHub, the recommended way of using `gadget3` is to use the versions released to CRAN. Defined releases make it possible to declare which versions of `gadget3` a model is compatible with, and tools such as `renv` (Ushey and Wickham, 2025) can be used to recreate a set of packages used, improving reproducibility.

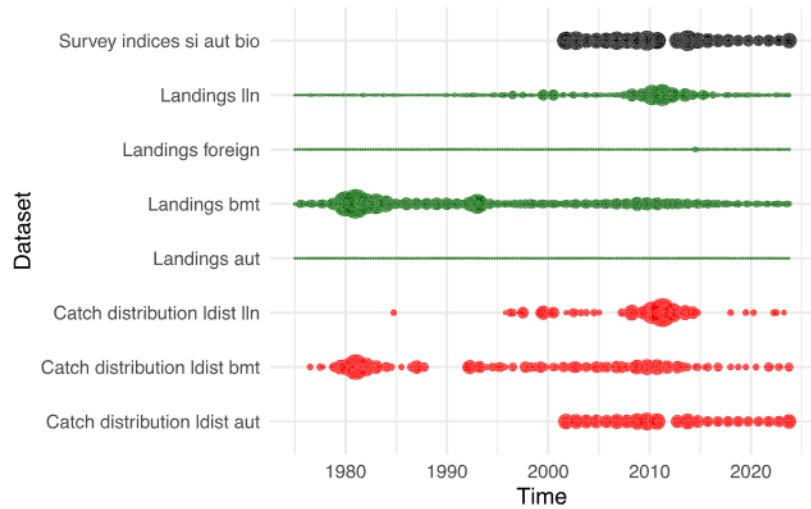


Fig. 5. Overview of the input data used for the blue ling model.

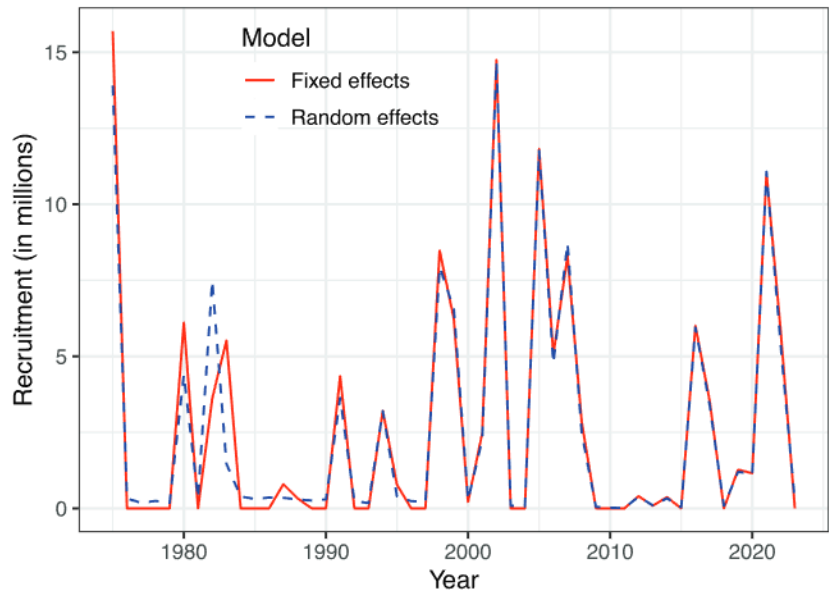


Fig. 6. Annual recruitment (age-2 individuals entering the immature stock).

Table 2
Summary of gadget2 greater silver smelt optimisation, per iterative reweighting stage.

Stage	Component	Function calls			Score	Time (s)
		Simulated annealing	Hooke & Jeeves	BFGS		
1	aldist.aut	434 160	2586	1920	212	3676
1	aldist.comm	506 520	2670	1920	2304	4232
1	ldist.aut	460 080	2649	1920	37	3877
1	ldist.comm	748 440	2673	1920	510	6052
1	matp.aut	408 240	2986	1920	1159	3417
1	survey.indices	581 040	2800	1920	21 349	4781
2	nodf	408 240	2524	1920	13	3119
2	final	451 440	2945	1920	10 597	3453

A CRAN release also gives some assurance of code quality. The CRAN repository will run gadget3 unit tests periodically, ensuring that gadget3 is compatible with other packages on CRAN. We also use GitHub Actions to run these tests on a matrix of R versions and operating systems, and CRAN also perform their own checks on a variety of architectures and operating systems.

8. Future work

gadget3 has achieved its initial aims at this stage, however there are various avenues of development to explore. Whilst multi-area support and tagging are both available and the components have been tested in isolation (e.g., `test-stock_areas.R`), there is yet to be a full end-to-end

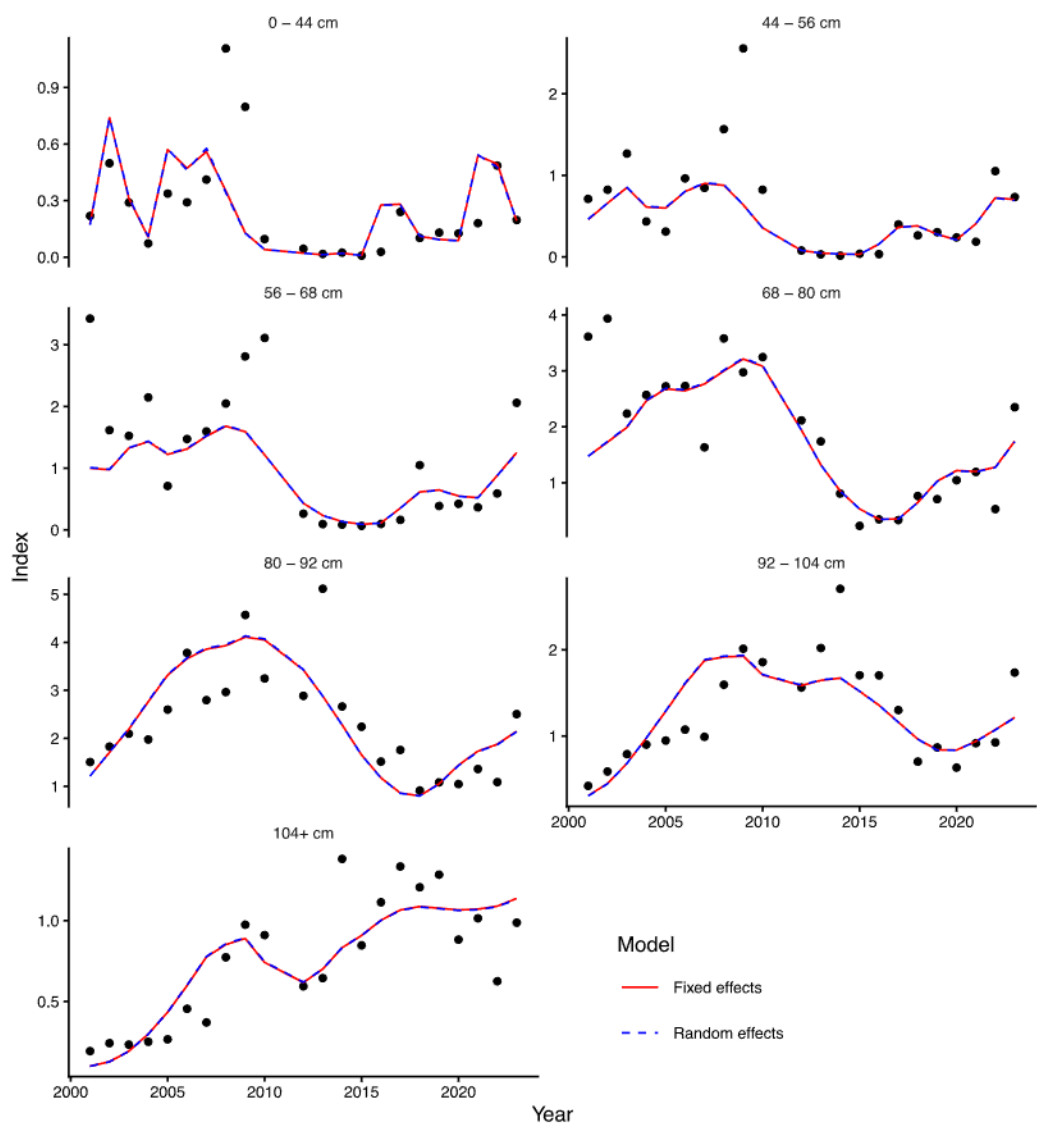


Fig. 7. Observed (points) versus model-predicted (solid line = fixed effects model, dashed line = random effects model) survey indices, disaggregated by length stratum. The model fits a log-linear relationship with slope fixed at 1.

Table 3
Summary of gadget3 greater silver smelt optimisation, per iterative reweighting stage. The final parameter values are taken from the stage 2 component that minimises the NLL, i.e., component 'ldist.aut' in this example.

Stage	Component	Function calls		Converged?	Score	Time (s)
		Function	Gradient			
1	aldist.aut	20 111	2000	FALSE	3032	1386
1	aldist.comm	10 461	1084	TRUE	1782	775
1	ldist.aut	20 711	2000	FALSE	526	1404
1	ldist.comm	15 965	1661	TRUE	1151	1138
1	survey.indices	14 262	1319	TRUE	27 283	985
1	matp.aut	19 769	1915	TRUE	1372	1330
2	aldist.aut	18 187	1798	TRUE	10 215	1178
2	aldist.comm	14 583	1462	TRUE	16 288	984
2	ldist.aut	10 784	1115	TRUE	10 215	739
2	ldist.comm	5944	630	TRUE	10 280	431
2	survey.indices	23 672	2000	FALSE	17 195	1493
2	matp.aut	8561	824	TRUE	10 306	589
NA	final	NA	NA	NA	10 215	NA

example developed using these facilities in a real world scenario. Doing so may reveal scaling issues that do not come up in small examples, which will need to be worked around.

One of the core assumptions copied from gadget2 into gadget3 is lengthgroup vectors—it is assumed that all stocks can be binned into a vector of length bins. Number in each age-length cell is distributed

based on Stefansson (2005). Whilst this is very flexible, it does not scale particularly well for stocks with wide-ranging sizes; either there is very little length resolution or the tape is weighed down with the quantity of calculations. Unevenly sized length bins would be one solution but the default growth formulation would cope poorly with this. Work is underway to allow the user to replace length bins with a smaller number of platoons (Taylor et al., 2021), or substocks, each with their own length distribution based only on mean length at age in contrast with the fixed age-length bins currently implemented in gadget3.

gadget3 also has an initial implementation of Close-kin mark-recapture (CKMR Bravington et al., 2016). However, as currently implemented it cannot cope with migration between areas. Given tagging is often used to gain insights into migration, this is not ideal. It is hoped that the platoons mentioned above simplify the tracing family lineages through model history, and full CKMR becomes possible.

Integrated analysis of fish population dynamics builds upon multiple disparate datasets through composite likelihoods (Maunder and Punt, 2013). Whilst conceptually simple there a number of practical issues when combining the likelihood components, namely the issue of data weighing (Francis, 2011, and related papers). Gadget models have in the past used inverse variance weighting based on the iterative reweighting heuristic (Stefansson, 2003; Taylor et al., 2004). This heuristic, as mentioned above, has also been implemented for gadget3, but with the inclusion of random effects in to gadget models it is now possible to better investigate the various source of uncertainty and biases than before (e.g. by following process suggested by Thorson et al. (2023)).

Standardised model diagnostic, made available with the combination of gadgetutils and gadgetplots, improve upon the communication between modellers and help to identify issues or avenues for future model improvements. This is inspired by the approaches taken by Taylor et al. (2021) and stockassessment.org, but suited to the types of output from gadget models. Future work on these fronts would entail incorporating one step ahead residuals (see Thygesen et al., 2017; Trijoulet et al., 2023) into standard gadget model diagnostics.

9. Conclusion

gadget3 is major update to the already flexible gadget2 framework (Begley and Howell, 2004). It allows modellers to implement a broad range of model structures including multi-stock, multi-area, multi-fleet dynamics by chaining together simple processes. Its strong modularity enhances model testing, customisation and development of new features making gadget3 a broadly applicable tool for fish and fisheries population dynamics modelling, and one of the few generic frameworks for the implementation of MICE models.

gadget3 aims to provide a rigorous quantitative and statistical framework to enhance question-driven implementations where simplicity and complexity can be added where necessary and integrated within the same model (e.g., multispecies implementations coupling size structure and biomass production models). We argue this could serve model implementations for supporting both tactical and strategic decision making. In this context, the possibility to implement alternative model structures and resolutions of the different system components, processes and data all within the same modelling framework is particularly attractive and well supported by dedicated data warehousing tools for ecosystem modelling such as MFDB (Lentin and Elvarsson, 2021). gadget3 continues to be under active development, processes and methods within are added to and refined regularly, the work being coordinated on GitHub.

Punt et al. (2020) lists a number of features for next generations assessment modelling framework. This list of features was used as a guide for the development of gadget3. Although not all features have been implemented, gadget3 framework is flexible enough to allow for the development of said features when need arise.

CRediT authorship contribution statement

Jamie Lentin: Writing – review & editing, Writing – original draft, Validation, Software, Conceptualization. **Bjarki Elvarsson:** Writing – review & editing, Writing – original draft, Validation, Funding acquisition, Conceptualization. **William Butler:** Writing – review & editing, Writing – original draft, Validation, Data curation. **Valerio Bartolino:** Writing – review & editing, Writing – original draft, Validation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

gadget3 has received funding from the Innviðasjóður Icelandic infrastructure grant #191774-0031, the Under One Cod project #206740-051, the NORSUSTAIN project funded by the joint Danish, Greenlandic, and Faroese Presidency of the Nordic Council of Ministers, the Formas Swedish Research Council for Sustainable Development #2021-00826.

Appendix A. Available gadget3 actions

Tables A.1 and A.2 describe the full list of actions available in gadget3, and where relevant the gadget2 userguide section to equivalent functionality.

Appendix B. Blue ling model description and results

The model spans the period from 1975 to 2023, with each year divided into four quarterly time-steps. Two sub-stocks are modelled: an immature component, comprising ages 2–15 years and lengths 20–112 cm, and a mature component, comprising ages 5–20 years and lengths 36–140 cm, with the oldest age treated as a plus group (≥ 20 years). In both components, length is partitioned into 8-cm bins. Individuals transfer from the immature to the mature component through two processes: maturation, governed by a length-based maturity ogive, and ageing, whereby immature fish reaching 15 years are transferred to the mature component at the end of each year. Recruitment enters the immature component at age 2 in the first time-step. Fish are removed from both components through natural mortality, assumed constant across all ages ($M = 0.15$), and through commercial exploitation.

The fundamental state variable is $N_{a,l,s,y,t}$: numbers of individuals belonging to stock s , length class l , age a , at year y and quarter t . The indices run over:

Symbol	Domain	Description
s	{imm, mat}	Immature and mature substocks
l_{imm}	{20, 28, ..., 116} cm	8 cm bins, 13 classes
l_{mat}	{36, 44, ..., 140} cm	8 cm bins, 14 classes
a_{imm}	{2, ..., 15} yr	Age in years
a_{mat}	{5, ..., 20} yr	Age in years (plus group at 20)
y	{1975, ..., 2023}	Year
t	{1, 2, 3, 4}	Quarter, $\Delta t = 0.25$ yr

Within each time step the following operations are applied in sequence: natural mortality $\rightarrow$ ageing $\rightarrow$ growth and maturation $\rightarrow$ recruitment $\rightarrow$ fishing.

Table A.1

Available gadget3 stock dynamics actions and their parallels from gadget2 userguide. [Begley \(2005\)](#).

Action function	Description	Userguide section
g3a_age	Ageing of stock, moving from one age group to next	4
g3a_growmature	Combined growth & maturation	4.3, 4.10
g3a_initialconditions	Defining initial conditions of stock	4.8, parameterisations from Þór Elvarsson et al. (2018) now built-in
g3a_mature	Maturation (when not tied to growth)	4.10
g3a_migrate	Migration of stock between areas	4.9
g3a_naturalmortality	Natural mortality of stock	4.5
g3a_otherfood	Reset a stock to a known state	6
g3a_predate	Predation, both by predators and fleets	4.7
g3a_predate_tagrelease	Tag & release the results of a predation step	5.1
g3a_renewal	Add extra individuals to stock	4.12, parameterisations from Þór Elvarsson et al. (2018) now built-in
g3a_spawn	Spawning (recruitment dependent on SSB abundance)	4.13, additional variant of Beverton–Holt from SS3 (Methot and Wetzel, 2013)
g3a_spmmodel	Apply surplus production model to stock	New to gadget3
g3a_tag_shedding	Remove tags from stock (i.e., move from tagged to untagged)	5.1
g3a_weightloss	Apply weight loss to a stock	New to gadget3

Table A.2

Available gadget3 likelihood actions and their parallels from gadget2 userguide [Begley \(2005\)](#).

Action function	Description	Userguide section
g3l_abundancedistribution	Compare stock abundance to observed data	8.6
g3l_catchdistribution	Compare catch to observed data	8.3, 8.4, 8.5, 8.7 8.13
g3l_bounds_penalty	Ensure parameters stay within given bounds	8.1
g3l_random_dnorm	Ensure parameter values fit a normal distribution	New to gadget3
g3l_random_walk	Ensure parameter values fit a random walk	New to gadget3
g3l_sparsesample	Compare a sparse set of observations, including derived values	New to gadget3
g3l_understocking	Ensure stocks cannot be fished over available stock	8.2

Population numbers are governed by the following equations:

$$\begin{aligned}
 N_{a,l,s,y,t+1} &= \sum_{l'} G_{l \rightarrow l'} \left[(N_{a,l',s,y,t} - \sum_f C_{f,a,l',s,t}) e^{-M_a \Delta t} + A_{a,l',l,s,y,t} \right] \\
 &\quad \text{if } t < 4 \\
 N_{a,l,s,y,t+1} &= \sum_{l'} G_{l \rightarrow l'} \left[(N_{a-1,l',s,y,t=4} - \sum_f C_{f,a-1,l',s,t=4}) e^{-M_{a-1} \Delta t} + A_{a-1,l',l,s,y,t=4} \right] \\
 &\quad \text{if } a_{s,min} < a < a_{s,max} \\
 N_{a,l,s,y,t+1} &= \sum_{l'} G_{l \rightarrow l'} \left[(N_{a,l',s,y,t=4} - \sum_f C_{f,a,l',s,y,t=4} + \right. \\
 &\quad \left. N_{a-1,l',s,y,t=4} - \sum_f C_{f,a-1,l',s,y,t=4}) e^{-M_a \Delta t} \right] \\
 &\quad \text{if } a = a_{max}
 \end{aligned}
 \tag{B.1}$$

where $G_{l \rightarrow l'}$ is the proportion in length-group l that has grown $l - l'$ length units (e.g., cm) in Δt . $C_{f,a,l',s,y,t}$ denotes the catches by fleet $f \in \{S, L, F, T\}$, i.e. the fleets representing the survey (S), longliners (L), foreign vessels (F), and trawlers (T). M_a is the natural mortality at age a and $A_{a,l',l,s,y,t}$ denotes the numbers of immature fish at length l' that have matured (i.e., “moved” from the immature to the mature stock component) as they grew to length l . Throughout the model description, l and l' are used either to reflect two separate length groups, or the midpoints of these length-group intervals, depending on the context.

B.1. Initial conditions

At $t = 0$ (start of 1975), numbers at age are initialised from an equilibrium abundance vector. Numbers-at-age decay with a combined natural and initial fishing mortality F_0 :

$$N_{s,a,0} = R_0 \cdot e^{-(a-a_r)(M+F_0)} \tag{B.2}$$

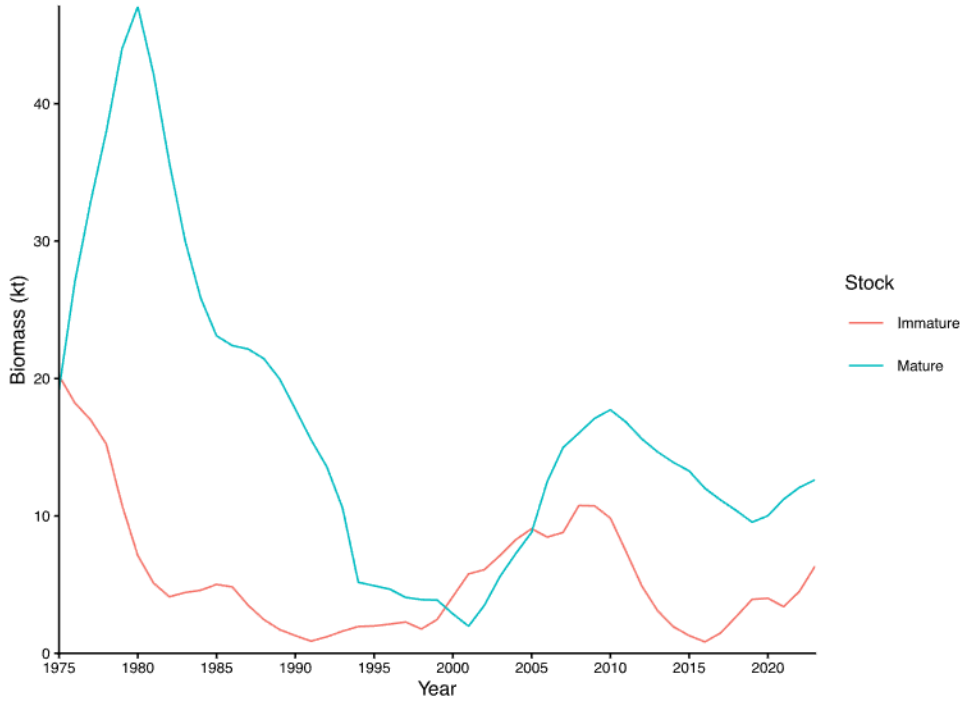


Fig. B.1. Total biomass (tonnes) for immature and mature substocks.

where $a_r = 2$ is the age at first recruitment and R_0 is the initial scalar (estimated). The partition between stocks uses a knife-edge rule at age 10: ages < 10 are assigned to the immature stock ($s = \text{imm}$), ages ≥ 10 to the mature stock ($s = \text{mat}$).

The length distribution within each age class is taken as normal with mean μ_a (from the von Bertalanffy growth function evaluated at that age) and standard deviation proportional to the mean:

$$N_{s,l,a,t} \propto N_{s,a,0} \cdot \phi\left(\frac{l - \mu_a}{\sigma_a}\right), \quad \sigma_a = cv_a \cdot \mu_a \quad (\text{B.3})$$

where ϕ is the standard normal density and cv_a is an age-specific coefficient of variation (fixed, not estimated).

B.2. Natural mortality

Continuous exponential mortality is applied within each quarterly step:

$$N'_{s,l,a,t} = N_{s,l,a,t} \cdot e^{-M_a \Delta t} \quad (\text{B.4})$$

Natural mortality $M_a = 0.15 \text{ yr}^{-1}$ is shared across stocks and ages and is not estimated, giving a per-quarter survival of $e^{-0.15 \times 0.25} \approx 0.963$.

B.3. Growth

B.3.1. Mean increment

The expected length increment per quarter follows the von Bertalanffy equation evaluated over $\Delta t = 0.25 \text{ yr}$:

$$\Delta \bar{l}(l) = (L_\infty - l)(1 - e^{-K \Delta t}) \quad (\text{B.5})$$

Growth parameters are shared between the two stocks and are fixed at externally estimated values ($L_\infty = 125.16 \text{ cm}$, $K = 0.1264 \text{ yr}^{-1}$, $t_0 = 0$) based on values from an adjacent stock.

B.3.2. Growth dispersion

Rather than advancing all individuals by their mean increment, growth is dispersed around the mean to length bins $\Delta l \in \{0, 1, \dots, n_{\max}\}$

that an individual advances per step, following a beta-binomial density:

$$G_{l \rightarrow l'} = B(\alpha, \beta, n) \quad (\text{B.6})$$

with a mean $\alpha\beta = \Delta l$ and α is subject to

$$\alpha = \frac{\beta \Delta \bar{l}}{n - \Delta \bar{l}}$$

with n fixed as the maximum length group steps over which a fish can grow in a particular timestep, and β estimated. Note that inherent in this approach is that the growth increment is always positive, which requires a slightly different interpretation of the k and L_∞ parameters compared to what one would expect from a more traditional Von Bertalanffy age-based model (i.e., L_∞ is rather like an upper boundary than a mean maximum expected length).

B.4. Maturation

At every time step, a fraction of the immature stock matures and transfers to the mature stock. Maturation is length-dependent and follows a continuous logistic ogive:

$$\psi(l) = \frac{1}{1 + e^{-\alpha_{\text{mat}}(l - l_{50,\text{mat}})}} \quad (\text{B.7})$$

where α_{mat} controls the steepness and $l_{50,\text{mat}}$ is the length at 50% maturity — both estimated. The number transferred each step is:

$$\Delta N_{\text{mat}} = \psi(l) \cdot N_{\text{imm},l,a,t} \quad (\text{B.8})$$

B.5. Recruitment

Recruits enter the immature stock annually at age $a_r = 2$. Annual recruitment R_y follows an optional log-AR(1) process:

$$\log R_y = \varphi \log R_{y-1} + (1 - \varphi) \log \bar{R} + \varepsilon_y, \quad \varepsilon_y \sim \mathcal{N}(0, \sigma_R^2) \quad (\text{B.9})$$

where φ is the autocorrelation coefficient, $\bar{R}$ is the long-run mean recruitment, and σ_R is the log-recruitment standard deviation — all estimated. Recruits are distributed over length classes at age a_r using the same normal-CV formulation as the initial conditions (Eq. (B.3)).

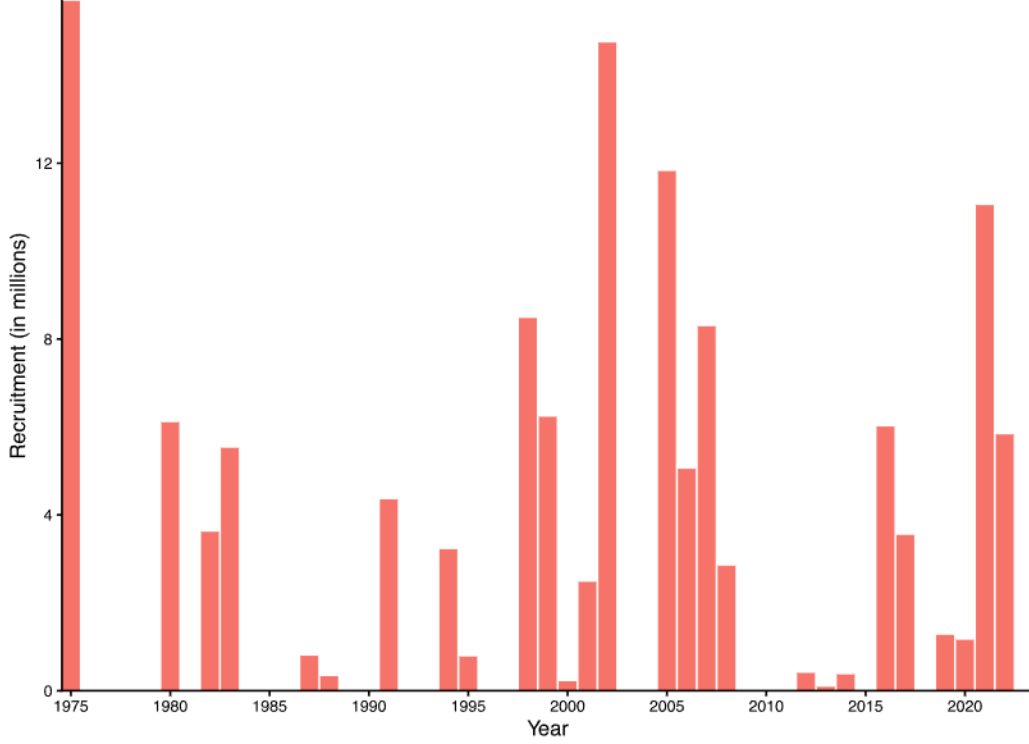


Fig. B.2. Annual recruitment (age-2 individuals entering the immature stock).

The log-AR(1) process is disabled for the main fixed effects model, but enabled for the random effects variant.

B.6. Fishing

Four fleets are included in the model: three commercial fleets, a bottom trawl fleet (combining Icelandic and Greenlandic waters), a longline fleet, and a fleet of foreign vessels, and one survey fleet representing the Icelandic autumn groundfish survey (2001–present). Each fleet is assigned a logistic selection curve, with the foreign fleet sharing the selection pattern of the longline fleet.

B.6.1. Fleet selectivity

Three commercial fleets (longline, bottom trawl, foreign) and one survey (autumn) each have a length-dependent selectivity function. The form is a logistic (exponential-L50) curve:

$$S_f(l) = \frac{1}{1 + e^{-\alpha_f(l-l_{50,f})}} \quad (\text{B.10})$$

Parameters α_f and $l_{50,f}$ are estimated per fleet. The foreign fleet shares the longline selectivity parameters, adding no additional degrees of freedom.

B.6.2. Catches and fishing mortality

Annual landings data were fleet- and time-step-specific ($L_{f,y,t}$), and assimilated into the model as direct removals of biomass by first simulating the expected composition of catches based on Eq. (B.11). The expected compositions of catches in biomass ($C_{f,a,l,s,y,t}^B$) were determined by multiplying landings ($L_{f,y,t}$) by the proportion of the total biomass available to be fished at a given age, length and stock component (immature/mature). The proportion was defined by biomass at a specific value of a , l , and s in the numerator, as available to the fleet (i.e., multiplied by selectivity S_f , divided by the sum of all biomass fishable by the fleet (i.e., values summed over the full range s' , l' , and

a' in the denominator). Catches in numbers are then defined by dividing by stock- and length-specific weights ($C_{f,a,l,s,y,t}$):

$$C_{f,a,l,s,y,t}^B = L_{f,y,t} \frac{S_f(l)N_{a,l,s,y,t}W_{l,s}}{\sum_{s'} \sum_{l'} \sum_{a'} S_f(l')N_{a',l',s',y,t}W_{l',s'}} \quad (\text{B.11})$$

$$C_{f,a,l,s,y,t} = \frac{C_{f,a,l,s,y,t}^B}{W_{l,s}}$$

B.6.3. Harvest control rule (projections)

When projecting beyond the historical period, observed landings are replaced by a hockey-stick harvest control rule applied annually to the mature-stock biomass (SSB):

$$HR(B) = h \cdot \min\left(\frac{B}{B_{\text{trigger}}}, 1\right) \quad (\text{B.12})$$

where h is the target harvest rate (fixed at the 2020 observed rate) and B_{trigger} is a biomass reference point. The resulting quota is divided equally among the three commercial fleets.

B.7. Likelihood and objective function

The model is fitted by minimising a weighted composite objective function:

$$\mathcal{L} = \sum_k w_k \ell_k \quad (\text{B.13})$$

B.7.1. Length distribution components

Applied to commercial fleet catch-at-length (longline, bottom trawl) and the autumn survey, including a maturity-proportion component that uses catch split by stock (imm/mat):

$$\ell_{\text{ldist},f} = \sum_l \sum_l (\hat{p}_{l,l,f} - p_{l,l,f})^2 \quad (\text{B.14})$$

where $p_{l,l,f}$ is the observed proportion at length l and $\hat{p}_{l,l,f}$ is the modelled proportion.

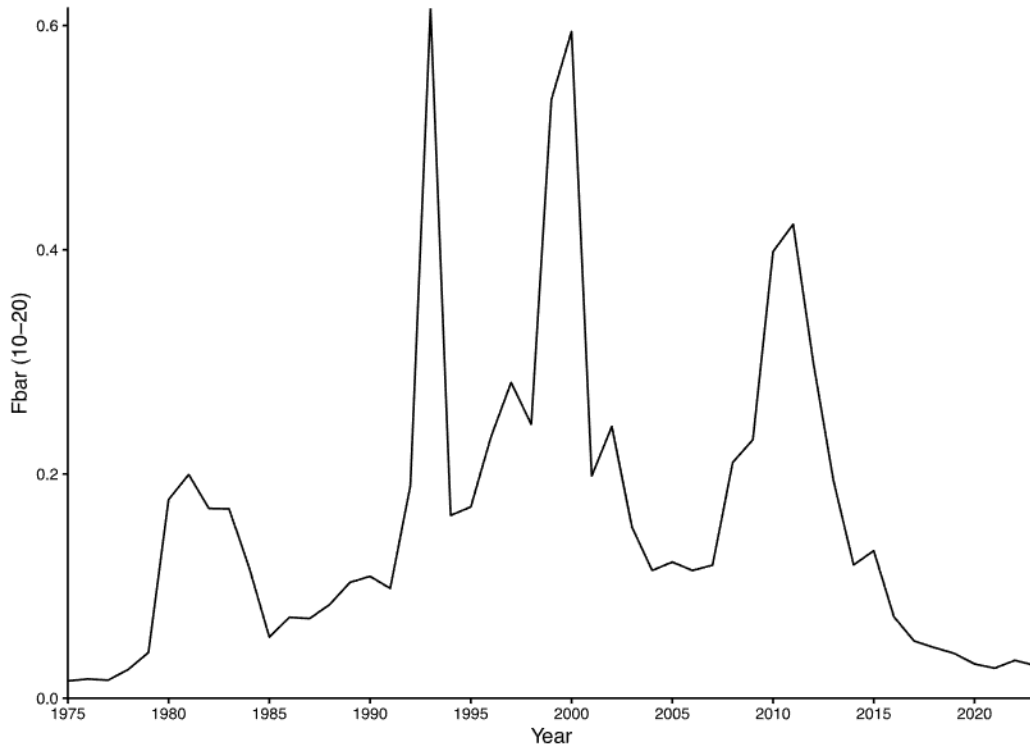


Fig. B.3. Estimated fishing mortality (ages 10–20).

B.7.2. Survey index components

Survey abundance indices are fitted with a log-linear model with slope $\beta = 1$ fixed, separately for each length stratum l :

$$\ell_{SL,l} = \sum_i (\log I_{l,i} - \log q_l - \log \hat{B}_{l,i})^2 \quad (\text{B.15})$$

where $I_{l,i}$ is the observed index, q_l is a stratum-specific catchability estimated analytically, and $\hat{B}_{l,i}$ is the modelled biomass in that stratum.

B.7.3. Understocking penalty

A large penalty (weight 100) is added if the model predicts negative stock numbers anywhere:

$$\ell_{\text{pen}} = \sum_{s,l,t} [\min(0, N_{s,l,t})]^2 \quad (\text{B.16})$$

B.7.4. Bounds penalty

A large penalty (weight 100) is added if the model predicts negative stock numbers anywhere:

$$\ell_{\text{pen}} = \sum_{s,l,t} [\min(0, N_{s,l,t})]^2 \quad (\text{B.17})$$

B.7.5. Iterative re-weighting

Weights, w_k , are assigned according to the iterative weighting heuristic introduced by Stefansson (2003), and subsequently implemented by Taylor et al. (2004). This heuristic is an inverse variance approach, $w_k \propto 1/\text{Var}(\text{residuals}_k)$, where the component variances are based on multiple model runs in which a particular component has been excessively emphasised. To avoid issues related to overfitting individual length-grouped survey indices, likelihood components were grouped and weighted together, thus ensuring that more than one survey index per year were available within a weighted group (see Tables B.2 and B.3).

B.8. Stock trajectories

B.8.1. Total biomass by stock

See Fig. B.1.

Table B.2

Summary of results of each optimisation stage and time taken, stage 1.

	Fn calls	Gd calls	Convergence	Score	Wallclock s
aut	13 469	1602	TRUE	1720.96	337.88
bmt	15 853	1978	TRUE	5388.37	398.89
ldist_lln	16 173	2046	TRUE	6436.28	416.48
matp_aut	20 958	2500	FALSE	978.46	523.98
sind	18 822	1694	TRUE	13 153.11	450.13

Table B.3

Summary of results of each optimisation stage and time taken, stage 2.

	Fn calls	Gd calls	Convergence	Score	Wallclock s
aut	6186	965	TRUE	4376.72	161.93
bmt	10 660	1598	TRUE	4378.28	280.84
ldist_lln	19 101	2500	FALSE	4386.56	487.74
matp_aut	7463	1275	TRUE	4378.25	200.51
sind	8791	1406	TRUE	4376.70	233.13

B.8.2. Recruitment

See Fig. B.2.

B.8.3. Fishing mortality

See Fig. B.3.

B.9. Survey index fits

Observed (points) versus model-predicted (line) survey indices, disaggregated by length stratum. The model fits a log-linear relationship with slope fixed at 1 See Fig. B.4.

B.10. Length distribution fits

B.10.1. Longline fleet

Fig. B.5.

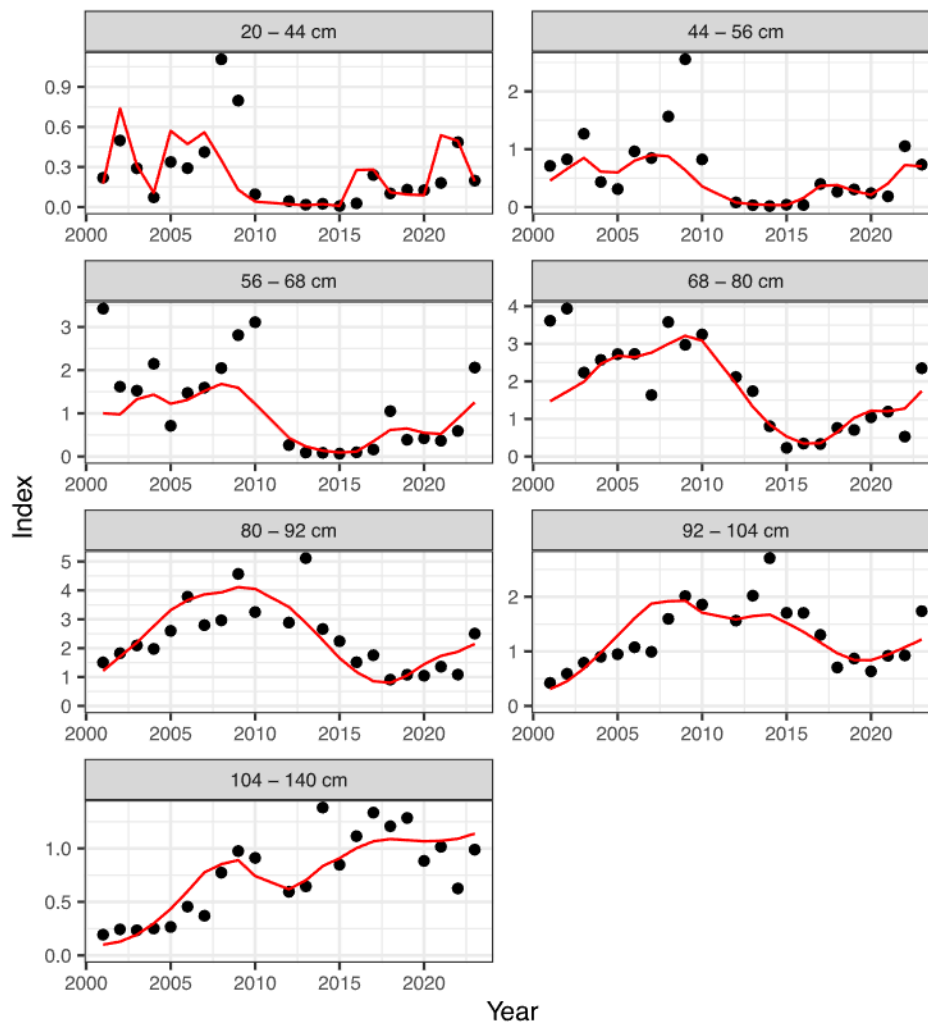


Fig. B.4. Survey index fits by length stratum.

B.10.2. Bottom trawl fleet

Fig. B.6.

B.10.3. Autumn survey

Fig. B.7.

B.11. Maturity proportion

Observed (points) versus predicted (line) proportions of mature fish at length from the autumn survey Fig. B.8.

B.12. Selectivity curves

Length-dependent selectivity for each fleet and the autumn survey Fig. B.9.

B.13. Likelihood scores

Contribution of each component to the total negative log-likelihood after iterative re-weighting Fig. B.10.

B.13.1. Residuals by component

Fig. B.11.

B.14. Catches

Total catches per stock Fig. B.12.

Appendix C. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.fishres.2026.107903>.

Data availability

All code is available on request, the GSS model uses commercial data not for distribution.

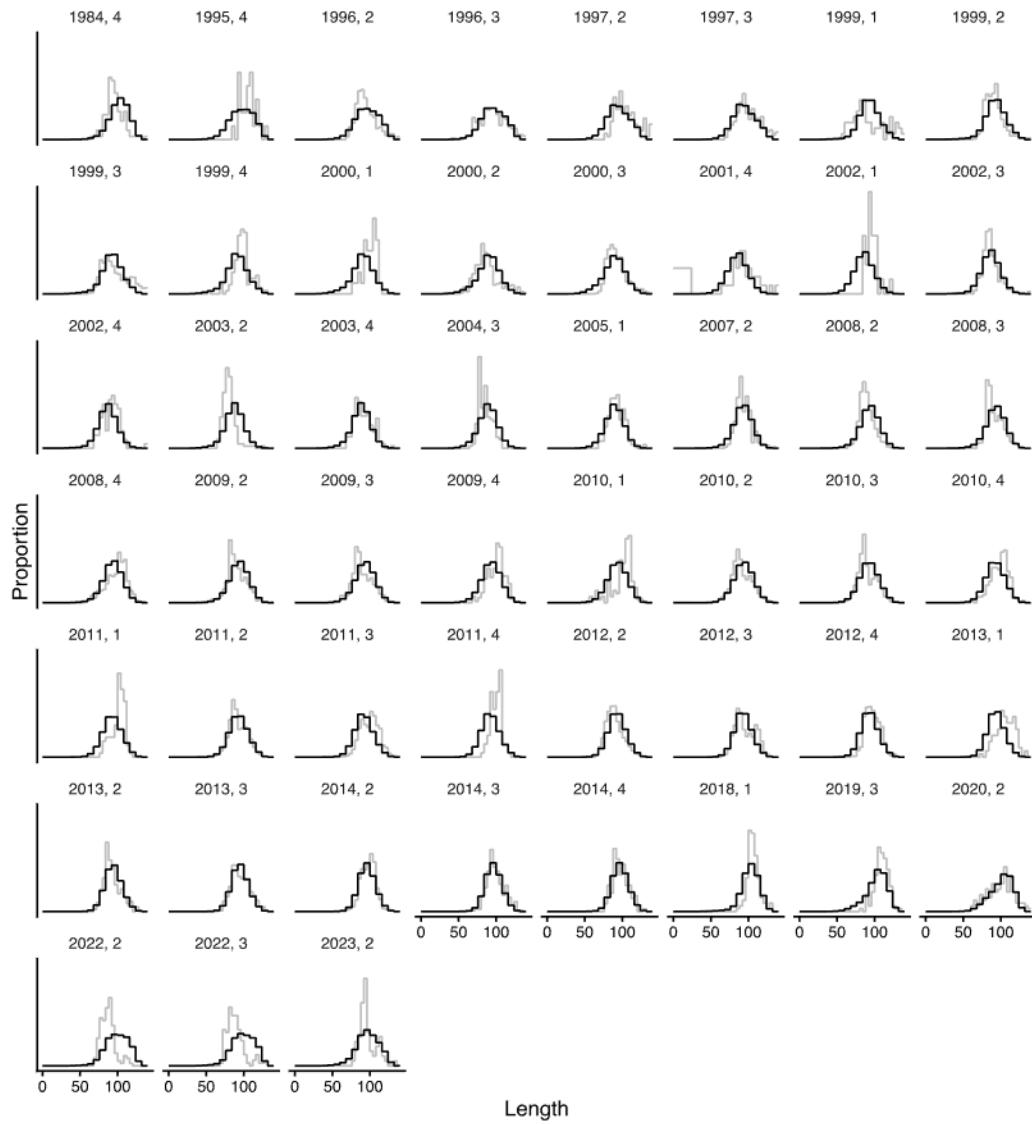


Fig. B.5. Observed (grey line) and predicted (black line) length distributions — longline fleet.

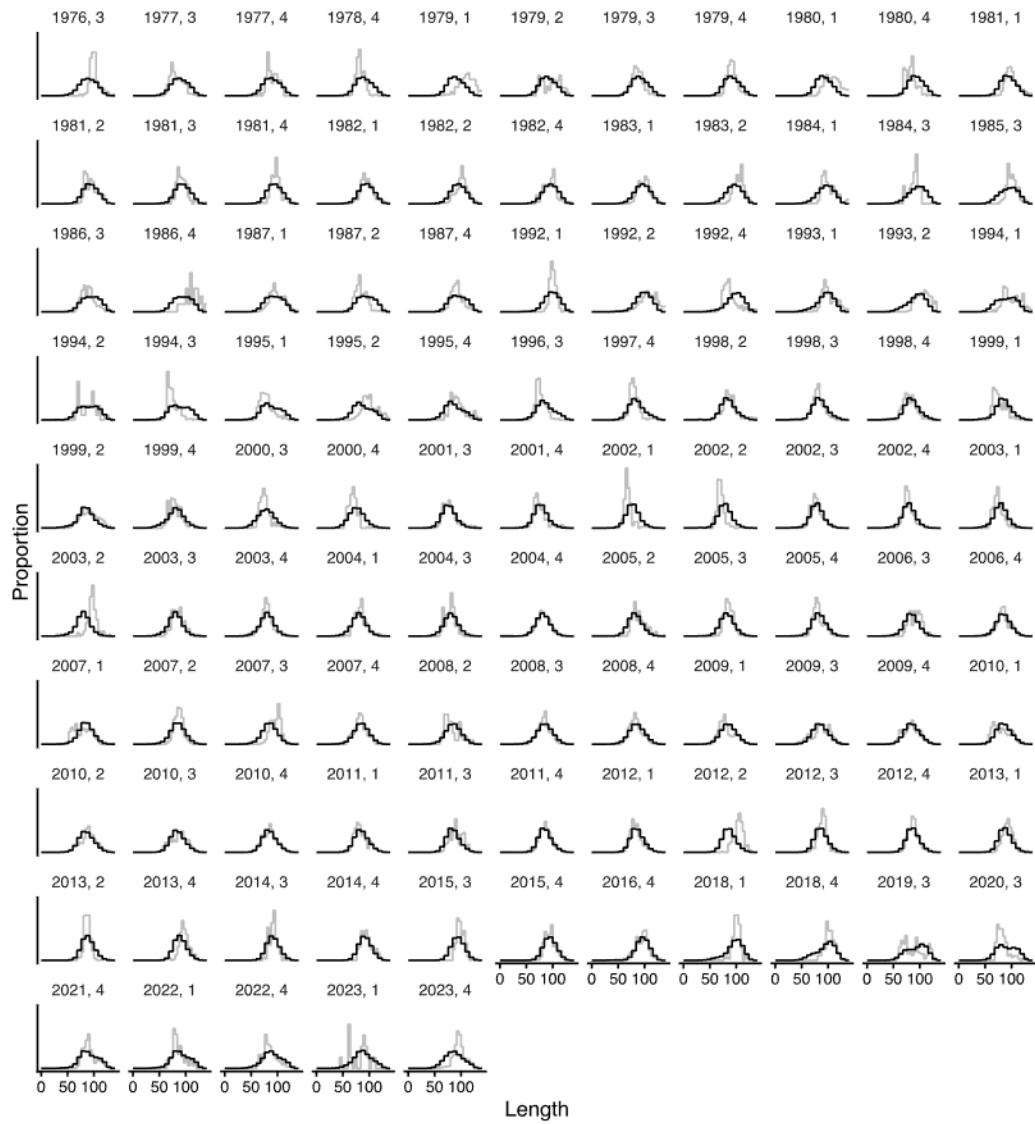


Fig. B.6. Observed (grey line) and predicted (black line) length distributions — bottom trawl fleet.

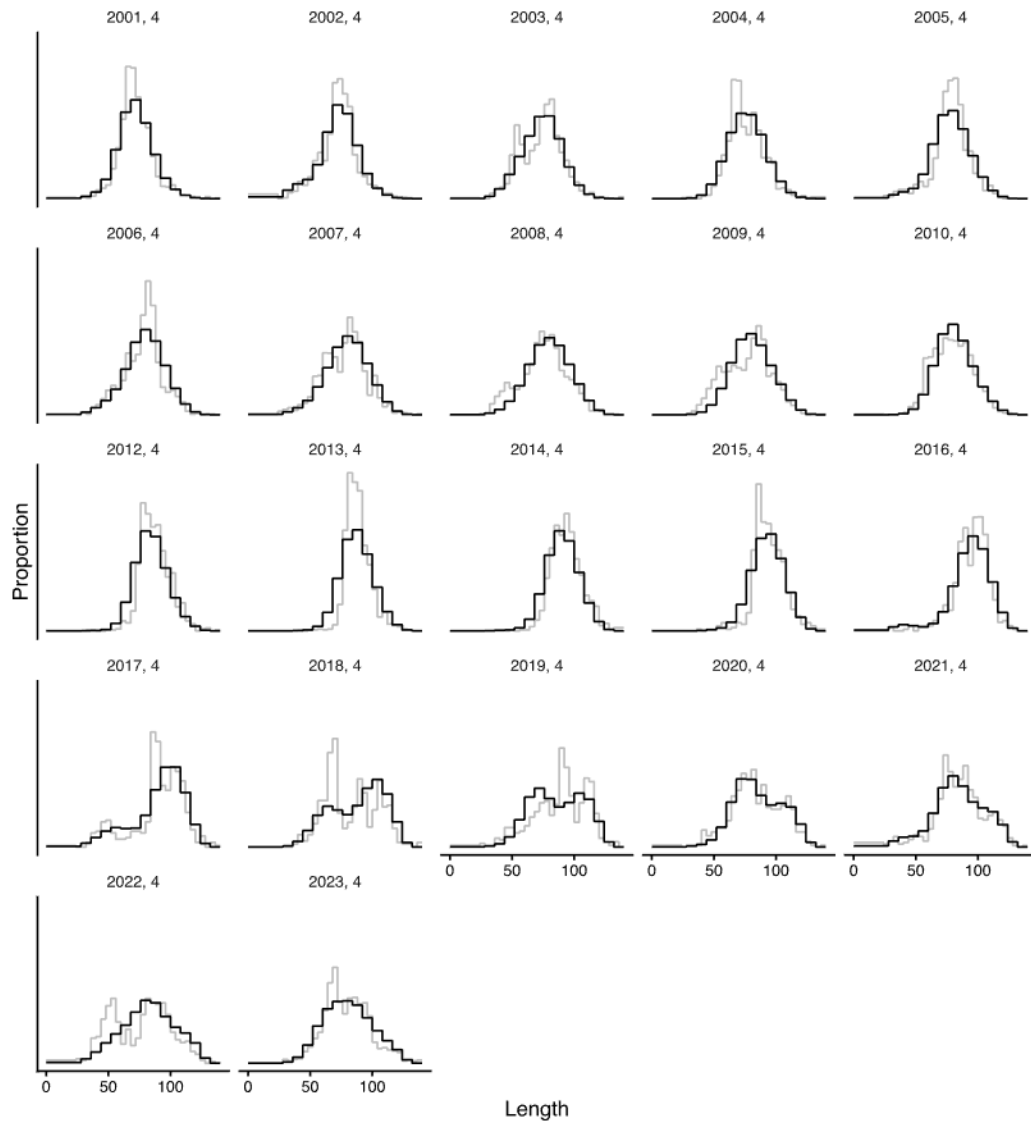


Fig. B.7. Observed (grey line) and predicted (black line) length distributions — autumn survey.

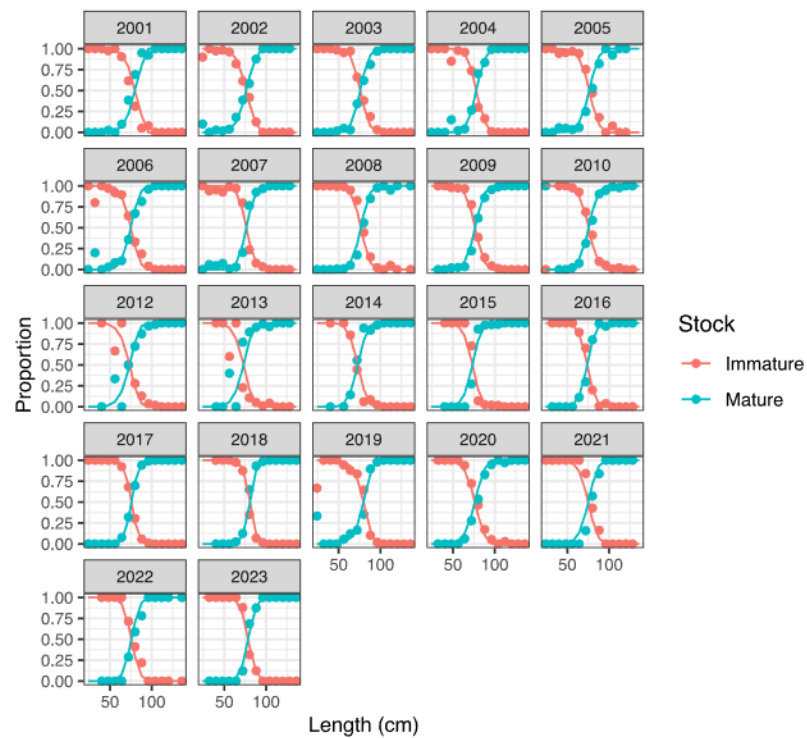


Fig. B.8. Maturity proportion by length — autumn survey.

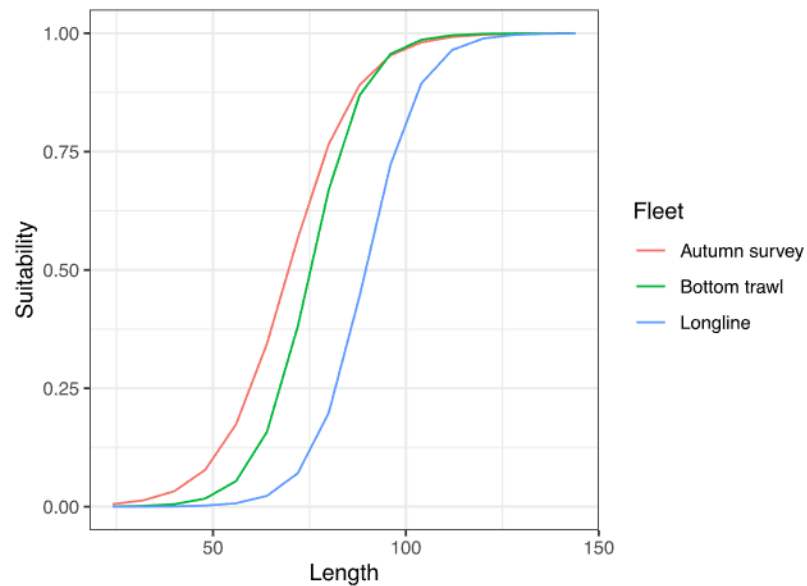


Fig. B.9. Estimated selectivity curves by fleet. Note that the foreign fleet shares the longline fleet's selection curve.

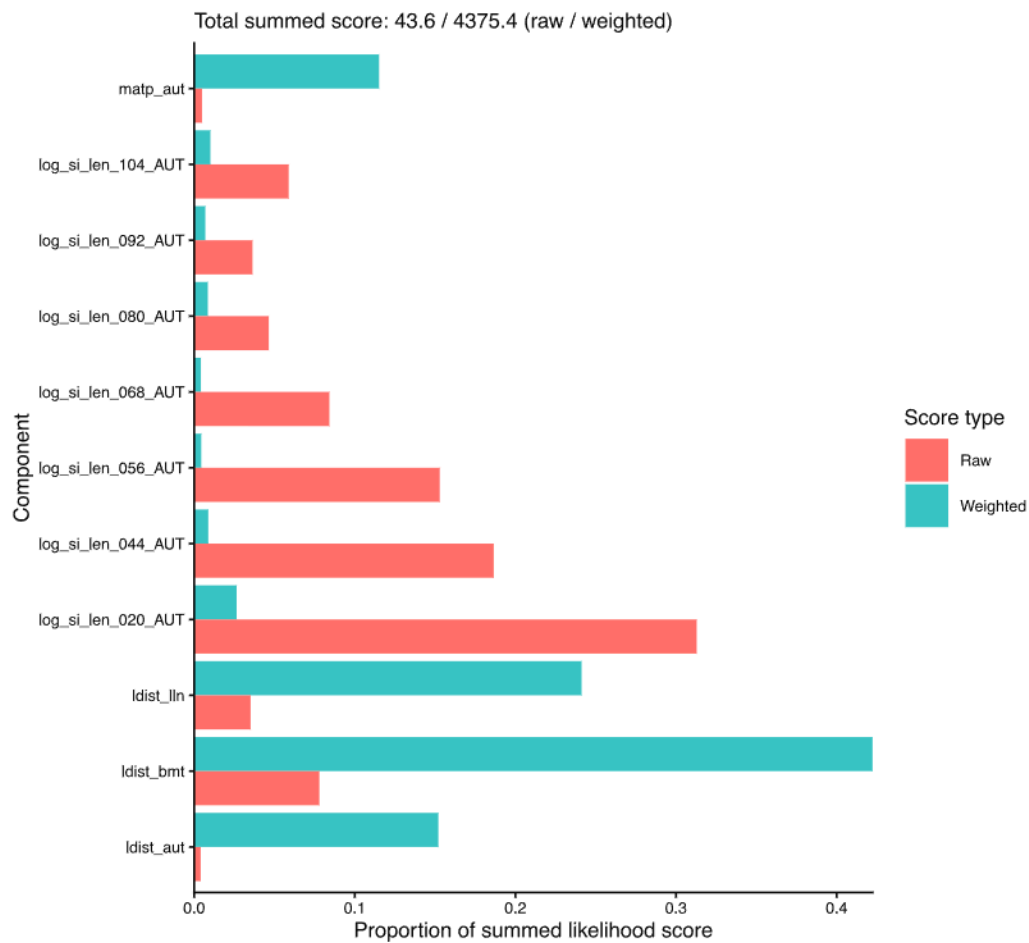


Fig. B.10. Weighted negative log-likelihood by component.

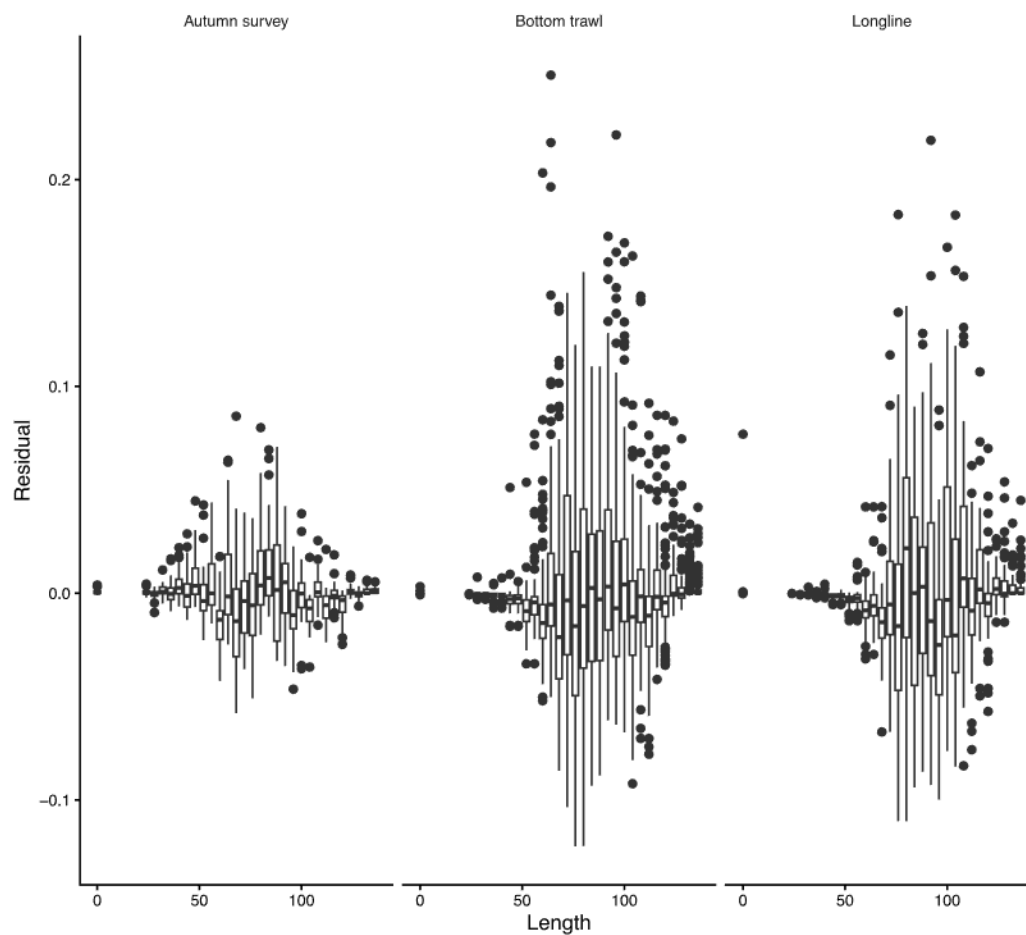


Fig. B.11. Length-based residuals for each catch composition likelihood component.

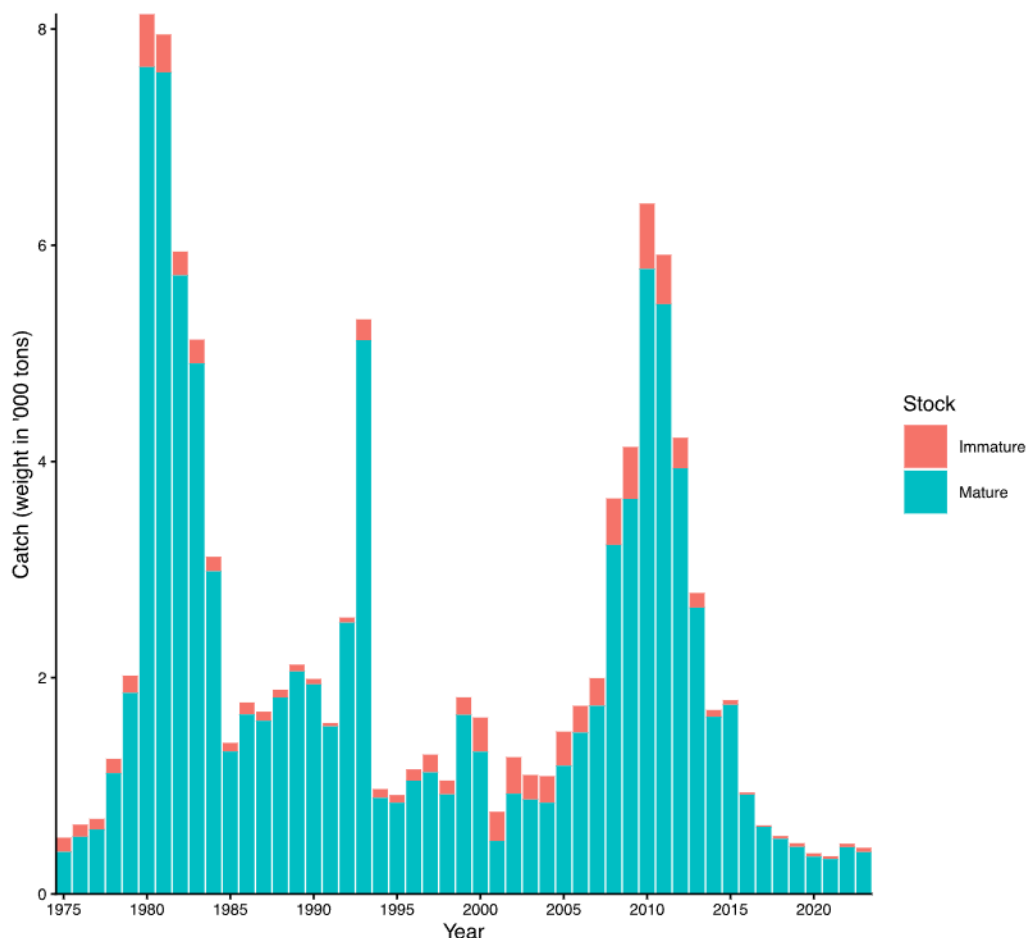


Fig. B.12. Annual catches per stock. Total catches are assumed known in the model.

References

- Adamczak, S.K., McHuron, E.A., Christiansen, F., Dunkin, R., McMahon, C.R., Noren, S., Pirotta, E., Rosen, D., Sumich, J., Costa, D.P., 2023. Growth in marine mammals: a review of growth patterns, composition and energy investment. *Conserv. Physiol.* 11, coad035. <http://dx.doi.org/10.1093/conphys/coad035>, arXiv:<https://academic.oup.com/conphys/article-pdf/11/1/coad035/50929660/coad035.pdf>.
- Ailloud, L.E., Hoenig, J.M., 2019. A general theory of age-length keys: combining the forward and inverse keys to estimate age composition from incomplete data. *ICES J. Mar. Sci.* 76, 1515–1523. <http://dx.doi.org/10.1093/icesjms/fsz072>, arXiv:<https://academic.oup.com/icesjms/article-pdf/76/6/1515/31247268/fsz072.pdf>.
- Begley, J., 2005. Gadget User Guide. Marine Research Institute, Reykjavík, Iceland, 90 Marine Research Institute Report 120.
- Begley, J., Howell, D., 2004. An overview of gadget, the globally applicable area-disaggregated general ecosystem toolbox. *ICES CM* 2004/FF 13.
- Bravington, M., Skaug, H., Anderson, E., 2016. Close-kin mark-recapture. *Statist. Sci.* 31, 259–274. <http://dx.doi.org/10.1214/16-ST552>.
- Butler, W., Elvarsson, B.T., Lentin, J., Vihtakari, M., 2025. Gadgetutils: Tools to create and maintain gadget3 models. URL: <https://github.com/gadget-framework/gadgetutils>. R package version 0.0.0.9000, commit a5dc90d53501db86d86d2f9c0b58b47210911cd5.
- Carpenter, B., Gelman, A., Hoffman, M.D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., Riddell, A., 2017. Stan: A probabilistic programming language. *J. Stat. Softw.* 76, 1–32. <http://dx.doi.org/10.18637/jss.v076.i01>, URL: <https://www.jstatsoft.org/index.php/jss/article/view/v076i01>.
- Cochran, W.G., 1977. Sampling techniques. John Wiley & sons.
- Elvarsson, B.T., Lentin, J., 2020. Rgadget: Tools to create and maintain gadget models. R package version 0.5.
- Fisheries, R., 1995. Code of conduct for responsible fisheries. 41.
- Fournier, D.A., Skaug, H.J., Ancheta, J., Iannelli, J., Magnusson, A., Maunder, M.N., Nielsen, A., Sibert, J., 2012. Ad model builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. *Optim. Methods Softw.* 27, 233–249. <http://dx.doi.org/10.1080/10556788.2011.597854>.
- Francis, R.C., 2011. Data weighting in statistical fisheries stock assessment models. *Can. J. Fish. Aquat. Sci.* 68, 1124–1138.
- Fríðriksson, Á., 1934. On the calculation of age-distribution within a stock of cod by means of relatively few age-determinations as a key to measurements on a large scale.
- Fulton, E.A., Smith, A.D., Johnson, C.R., 2004. Effects of spatial resolution on the performance and interpretation of marine ecosystem models. *Ecol. Model.* 176, 27–42. <http://dx.doi.org/10.1016/j.ecolmodel.2003.10.026>, URL: <https://www.sciencedirect.com/science/article/pii/S0304380003005416>.
- Gudmundsson, G., 1994. Time series analysis of catch-at-age observations. *J. R. Stat. Soc. Ser. C. Appl. Stat.* 43, 117–126. <http://dx.doi.org/10.2307/2986116>.
- Guldbrandsen Frøysa, K., Bogstad, B., Skagen, D.W., 2002. Fleksibest-an age-length structured fish stock assessment model. *Fish. Res.* 55, 87–101.
- ICES, 2019. ICES advice basis. <http://dx.doi.org/10.17895/ices.advice.5757>, URL: https://ices-library.figshare.com/articles/report/ICES_Advice_basis/18636095.
- ICES, 2023. Benchmark workshop on greenland halibut and redfish stocks (WKBNORTH). <http://dx.doi.org/10.17895/ices.pub.22304638.v1>, URL: https://ices-library.figshare.com/articles/report/Benchmark_workshop_on_Greenland_halibut_and_redfish_stocks_WKBNORTH/22304638.
- ICES, 2024. Benchmark workshop 3 on the development of MSY advice using spict (WKBMSYSPICT3). <http://dx.doi.org/10.17895/ices.pub.24998858.v1>, URL: https://ices-library.figshare.com/articles/report/Benchmark_workshop_3_on_the_development_of_MS_Y_advice_using_SPIC_T_WKBMSYSPICT3/24998858.
- ICES, 2025. Benchmark workshop on selected deep-sea fisheries stocks (wkb-deep). *ICES Sci. Rep.* 7:24, 148. <http://dx.doi.org/10.17895/ices.pub.28882295.v2>, URL: https://ices-library.figshare.com/articles/report/Benchmark_Workshop_on_Selected_Deep-Sea_Fisheries_Stocks_WKBDEEP/28882295.
- Kristensen, K., Nielsen, A., Berg, C.W., Skaug, H., Bell, B.M., 2016. TMB: Automatic differentiation and Laplace approximation. *J. Stat. Softw.* 70, 1–21. <http://dx.doi.org/10.18637/jss.v070.i05>.
- Kulatska, N., Neuenfeldt, S., Beier, U., Þór Elvarsson, B., Wennhage, H., Stefansson, G., Bartolino, V., 2019. Understanding ontogenetic and temporal variability of eastern baltic cod diet using a multispecies model and stomach data. *Fish. Res.* 211, 338–349. <http://dx.doi.org/10.1016/j.fishres.2018.11.023>, URL: <https://www.sciencedirect.com/science/article/pii/S0165783618303370>.
- Lentin, J., Elvarsson, B.T., 2021. Mfdb: MareFrame DB querying library. R package version 7.1-1.

- Lentin, J., Elvarsson, B.T., 2022. Gadget2to3: Translate G2 model files into G3. r package version 0.1.999, commit 2cbd6c905b4afb1ac65e46029ec3abf1f533d503. URL: <https://github.com/gadget-framework/gadget2to3>.
- Lindström, U., Smout, S., Howell, D., Bogstad, B., 2009. Modelling multi-species interactions in the barents sea ecosystem with special emphasis on minke whales and their interactions with cod, herring and capelin. *Deep. Sea Res. Part II: Top. Stud. Ocean.* 56, 2068–2079. <http://dx.doi.org/10.1016/j.dsr2.2008.11.017>, The Proceedings of the ECONORTH Symposium on Ecosystem Dynamics in the Norwegian Sea and Barents Sea. URL: <https://www.sciencedirect.com/science/article/pii/S0967064508003950>.
- Maunder, M.N., Punt, A.E., 2013. A review of integrated analysis in fisheries stock assessment. *Fish. Res.* 142, 61–74. <http://dx.doi.org/10.1016/j.fishres.2012.07.025>, URL: <https://www.sciencedirect.com/science/article/pii/S0165783612002627>.
- Methot, R.D., Wetzel, C.R., 2013. Stock synthesis: A biological and statistical framework for fish stock assessment and fishery management. *Fish. Res.* 142, 86–99. <http://dx.doi.org/10.1016/j.fishres.2012.10.012>, URL: <https://www.sciencedirect.com/science/article/pii/S0165783612003293>.
- Miller, T.J., Legault, C.M., 2015. Technical details for asap version 4.
- Nielsen, A., Berg, C.W., 2014. Estimation of time-varying selectivity in stock assessments using state-space models. *Fish. Res.* 158, 96–101. <http://dx.doi.org/10.1016/j.fishres.2014.01.014>, sI: Selectivity. URL: <https://www.sciencedirect.com/science/article/pii/S0165783614000228>.
- Pedersen, M.W., Berg, C.W., 2017. A stochastic surplus production model in continuous time. *Fish. Fish.* 18, 226–243.
- Pérez-Rodríguez, A., Howell, D., Casas, M., Saborido-Rey, F., Ávila-de Melo, A., 2017. Dynamic of the flemish cap commercial stocks: use of a gadget multispecies model to determine the relevance and synergies among predation, recruitment, and fishing. *Can. J. Fish. Aquat. Sci.* 74, 582–597. <http://dx.doi.org/10.1139/cjfas-2016-0111>.
- Pirotta, E., Mangel, M., Costa, D.P., Mate, B., Goldbogen, J.A., Palacios, D.M., Hückstädt, L.A., McHuron, E.A., Schwarz, L., New, L., 2018. A dynamic state model of migratory behavior and physiology to assess the consequences of environmental variation and anthropogenic disturbance on marine vertebrates. *Amer. Nat.* 191, E40–E56. <http://dx.doi.org/10.1086/695135>.
- Plagányi, E.E., Punt, A.E., Hillary, R., Morello, E.B., Thébaud, O., Hutton, T., Pillans, R.D., Thorson, J.T., Fulton, E.A., Smith, A.D.M., Smith, F., Bayliss, P., Haywood, M., Lyne, V., Rothlisberg, P.C., 2014. Multispecies fisheries management and conservation: tactical applications using models of intermediate complexity. *Fish. Fish.* 15, 1–22. <http://dx.doi.org/10.1111/j.1467-2979.2012.00488.x>, arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-2979.2012.00488.x> URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-2979.2012.00488.x>.
- Punt, A.E., Dunn, A., Þór Elvarsson, B., Hampton, J., Hoyle, S.D., Maunder, M.N., Methot, R.D., Nielsen, A., 2020. Essential features of the next-generation integrated fisheries stock assessment package: A perspective. *Fish. Res.* 229, 105617. <http://dx.doi.org/10.1016/j.fishres.2020.105617>, URL: <https://www.sciencedirect.com/science/article/pii/S016578362030134X>.
- Ryg, M., Lydersen, C., Markussen, N.H., Smith, T.G., Øritsland, N.A., 1990. Estimating the blubber content of phocid seals. *Can. J. Fish. Aquat. Sci.* 47, 1223–1227. <http://dx.doi.org/10.1139/f90-142>.
- Stefansson, G., 2003. Issues in multispecies models. *Nat. Resour. Model.* 16, 415–437.
- Stefansson, G., 2005. Growth models in population simulations. Final Report DST2: Development of Structurally Detailed Statistically Testable Models of Marine Populations, Technical Report, 118, pp. 73–80.
- Stefánsson, G., Pálsson, Ó., et al., 1997. Bormicon: a boreal migration and consumption model.
- Taylor, L., 2005. Datawarehouse for icelandic waters. Final Report: Dst²: Development of Structurally Detailed Statistically Testable Models of Marine Populations, Technical Report, volume 119, Marine Research Institute, Reykjavik.
- Taylor, L., Begley, J., Kupca, V., Stefansson, G., 2004. A simple implementation of gadget for cod in icelandic waters.
- Taylor, I.G., Doering, K.L., Johnson, K.F., Wetzel, C.R., Stewart, I.J., 2021. Beyond visualizing catch-at-age models: Lessons learned from the r4ss package about software to support stock assessments. *Fish. Res.* 239, 105924. <http://dx.doi.org/10.1016/j.fishres.2021.105924>, URL: <https://www.sciencedirect.com/science/article/pii/S0165783621000527>.
- Team, R. Core, 2025. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL: <https://www.R-project.org/>.
- Þór Elvarsson, B., Woods, P.J., Björnsson, H., Lentin, J., Thordarson, G., 2018. Pushing the limits of a data challenged stock: A size- and age-structured assessment of ling (molva molva) in icelandic waters using gadget. *Fish. Res.* 207, 95–109. <http://dx.doi.org/10.1016/j.fishres.2018.06.005>, URL: <https://www.sciencedirect.com/science/article/pii/S0165783618301759>.
- Thorson, J.T., Monnahan, C.C., Hulson, P.J.F., 2023. Data weighting: An iterative process linking surveys, data synthesis, and population models to evaluate mis-specification. *Fish. Res.* 266, 106762. <http://dx.doi.org/10.1016/j.fishres.2023.106762>, URL: <https://www.sciencedirect.com/science/article/pii/S0165783623001558>.
- Thygesen, U.H., Albertsen, C.M., Berg, C.W., Kristensen, K., Nielsen, A., 2017. Validation of ecological state space models using the laplace approximation. *Environ. Ecol. Stat.* 24, 317–339. <http://dx.doi.org/10.1007/s10651-017-0372-4>.
- Trijoulet, V., Albertsen, C.M., Kristensen, K., Legault, C.M., Miller, T.J., Nielsen, A., 2023. Model validation for compositional data in stock assessment models: Calculating residuals with correct properties. *Fish. Res.* 257, 106487. <http://dx.doi.org/10.1016/j.fishres.2022.106487>, URL: <https://www.sciencedirect.com/science/article/pii/S0165783622002648>.
- UN, 2002. World summit on sustainable development. Johannesburg. South Afr. 9, 67–69.
- Ushey, K., Wickham, H., 2025. Renv: Project environments. <http://dx.doi.org/10.32614/CRAN.package.renv>, r package version 1.1.4. URL: <https://CRAN.R-project.org/package=renv>.
- Vihitakari, M., Elvarsson, B.T., Butler, W., Lentin, J., 2025. Gadgetplots: Plot gadget3 models. r package version 0.3.1, commit 4ee07985b31a29f285adb8ec03698fb3daf7c61. URL: <https://github.com/gadget-framework/gadgetplots>.