



Pacific
Community
Communauté
du Pacifique



SEAPODYM

***Spatial Ecosystem and Population DYnamics Model
for Migratory Age-Structured Stocks***

Inna Senina, Olivier Titaud, Guillaume Briand, and Patrick Lehodey

—VERSION 4.0—

March 2022

Copyright ©2022 by SPC¹, PFRP² and CLS³.

LICENSE

THE PROGRAM (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS LICENSE. THE PROGRAM IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE PROGRAM OTHER THAN AS AUTHORIZED UNDER THIS LICENSE IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE PROGRAM PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Derivative Program" means a program based upon the Program or upon the Program and other pre-existing programs in which the Program may be recast, transformed, or adapted, except that a program that constitutes a Package will not be considered a Derivative Program for the purpose of this License.

b. "Licensor" means the individual or entity that offers the Program under the terms of this License.

c. "Original Author" means the individual or entity who created the Program.

d. "Package" means a software package, in which the Program in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent programs in themselves, are assembled into a collective package. Such a Package will not be considered a Derivative Program for the purposes of this License.

e. "Program" means the copyrightable computer software, including executable and source code, offered under the terms of this License.

f. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Program, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby

¹Oceanic Fisheries Programme, Secretariat of the Pacific Community, BPA5 Noumea, New Caledonia

²Pelagic Fisheries Research Program, University of Hawaii at Manoa. 1000 Pope Road, Honolulu, HI 96822 United States

³Marine Ecosystems Modelling, CLS, Sustainable Fisheries Management Business Unit. 11 rue Hermes, 31520 Ramonville Saint-Agne, France

grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Program as stated below:

a. to reproduce the Program, to incorporate the Program into one or more Packages, and to reproduce the Program as incorporated in the Packages;

b. to distribute copies of the Program including as incorporated in Packages;

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may distribute the Program only under the terms of this License, and You must include a copy of this License with every copy of the Program You distribute. You may not offer or impose any terms on the Program that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Program. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute the Program with any technological measures that control access or use of the Program in a manner inconsistent with the terms of this License Agreement. The above applies to the Program as incorporated in a Package, but this does not require the Package apart from the Program itself to be made subject to the terms of this License.

b. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation.

c. If you distribute the Program or any Packages, You must keep intact all copyright notices for the Program and give the Original Author credit reasonable to the medium or means You are utilizing by conveying the name (or pseudonym if applicable) of the Original Author if supplied; the title of the Program if supplied. Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Package, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

a. By offering the Program for public release under this License, Licensor represents and warrants that, to the best of Licensor's knowledge after reasonable inquiry:

i. Licensor has secured all rights in the Program necessary to grant the license rights hereunder and to permit the lawful exercise of the rights granted hereunder without You having any obligation to pay any royalties, compulsory license fees, residuals or any other payments;

ii. The Program does not infringe the copyright, trademark, publicity rights, common law rights or any other right of any third party or constitute defamation, invasion of privacy or other tortious injury to any third party.

b. EXCEPT AS EXPRESSLY STATED IN THIS LICENSE OR OTHERWISE AGREED

IN WRITING OR REQUIRED BY APPLICABLE LAW, THE PROGRAM IS LICENSED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES REGARDING FITNESS OF THE WORK FOR A PARTICULAR PURPOSE.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, AND EXCEPT FOR DAMAGES ARISING FROM LIABILITY TO A THIRD PARTY RESULTING FROM BREACH OF THE WARRANTIES IN SECTION 5, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE PROGRAM, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Packages from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Program). Notwithstanding the above, Licensor reserves the right to release the Program under different license terms or to stop distributing the Program at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You distribute the Program or a Package, the Licensor offers to the recipient a license to the Program on the same terms and conditions as the license granted to You under this License.

b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

d. This License constitutes the entire agreement between the parties with respect to the Program licensed here. There are no understandings, agreements or representations with respect to the Program not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Preface

SEAPODYM is a numerical modelling framework developed initially for investigating population dynamics of tunas under the influence of environment and fishing (www.spc.int/ofp/seapodym; www.seapodym.eu). The modelling effort started in 1995 at the Pacific Community (SPC) in Noumea, New Caledonia, under two consecutive EU-funded projects: SPR-TRAMP (1995-2000) and PROCFISH (2002-2005). The model development has continued under a grant from the Pelagic Fisheries Research Program (PFRP) at the University of Hawaii, allowing the implementation of a quantitative approach to estimate model parameters from massive spatio-temporal catch and size data (2004-2007). Since 2007 the development has been conducted at Collecte Localisation Satellites in Toulouse, France, in collaboration with the Oceanic Fisheries Program of SPC and under various research projects and funding provided by PFRP, the European Commission, Global Environment Facility (GEF) and the French National Research Agency.

SEAPODYM includes two conceptually different dynamical models: i) a model of spatio-temporal dynamics of functional groups of zooplankton and micronekton, that is called SEAPODYM for Lower and Mid-Trophic Level (SEAPODYM-LMTL), and ii) a model describing full spatio-temporal dynamics of a single species population, called SEAPODYM for Migratory Age-Structured Stocks (SEAPODYM-MASS). The underlying continuous equations of SEAPODYM-MASS are classical advection-diffusion-reaction equations with an ageing term, describing the population dynamics in time, age and two-dimensional space. Using environmental data and outputs of the SEAPODYM-LMTL model as a forcing, this model predicts the biomass of modelled species as well as the catch and size frequency of catch based on parameter estimation and data assimilation techniques.

This reference manual describes version 4.0 of the SEAPODYM-MASS model and software, which has been updated and enhanced with a method for integrating tagging data to inform movement parameters, a new mechanism of modelling seasonal migrations between feeding and spawning grounds, revised definitions of habitat indices and movement rates, and a method to account for fishing mortality and to predict catch in the absence of fishing effort data. In addition to a complete model, this version allows running the simulations and estimations of movement parameters of a single or selected cohorts using tagging data and building the following minimal models with maximum likelihood estimation (MLE) method: i) a model of spawning habitat, ii) a model of feeding habitats for selected age classes, and

iii) a model of virgin stock (without fisheries) with population density considered as an observation in the likelihood. These model applications can be used in model downscaling and/or adaptation to a different forcing dataset.

After a general introduction, this reference manual provides the full mathematical description of underlying continuous models, the numerical approximation scheme, a comprehensive guide on how to run numerical simulations and design various simulation studies, and a suite of numerical and statistical recipes for *data assimilation* in complex spatio-temporal fisheries models, aimed at informing model parameters from massive geo-referenced data and providing quantitative model reconstruction of observed phenomena as well as unbiased model predictions of the independent observations.

List of changes from version 3.0

- Integration of a movement model of tagged fish and tagging data likelihood.
- Implementation of an alternative approach to account for fishing mortality and to predict catch without fishing effort, i.e. based on observed catch only.
- Introduction of an additional likelihood term to allow estimation of minimal biomass supporting observed catches.
- A fully revised method to describe seasonal spawning migrations with a new mechanism of modelling large-scale migrations in the Eulerian model context.
- Revised spawning habitat as a product of prey, predator and thermal functions, the latter with seasonal variable when playing the role of movement habitat of spawning adults.
- Estimation of one additional parameter associated to each functional group of prey, providing more flexibility in the representation of vertical behaviour and access to preferred forage when computing the feeding habitat index.
- Decoupled optimal temperature and thermal range for larval survival and for feeding habitat at age zero, to allow a separate estimation of these parameters.
- Extension of sensitivity analysis by allowing two types of model global sensitivity runs: all-at-a-time (AAT) and one-at-a-time (OAT) runs, with names referring to the model parameter variations.
- A standalone habitat models build **seapodym_habitats**, computing either spawning or multiple feeding habitats (at selected ages) in a simulation or optimisation mode, with associated parameter estimation from provided habitat fields.
- A standalone model build **seapodym_densities** for population dynamics without fisheries, where biomass is considered both as observations and predictions in the likelihood.

- Various code modifications to minimise the memory use during adjoint gradient computation.

Contents

Introduction	13
1 The fish population dynamics model	23
1.1 Underlying equations	23
1.2 Dynamic processes	24
1.2.1 Species environment	25
1.2.2 Spawning habitat	27
1.2.3 Reproduction	29
1.2.4 Growth and maturity	31
1.2.5 Feeding habitat and movement rates	31
1.2.6 Natural mortality	43
1.2.7 Fishing mortality	45
1.3 Model parameters	48
References	50
2 Discretisation and numerical solution	53
2.1 Discretisation of model dimensions	53
2.1.1 Space discretisation	53
2.1.2 Time discretisation	54
2.1.3 Age discretisation	54
2.2 Approximation of equations and numerical scheme	55
2.2.1 Age integration	56
2.2.2 Time integration	56
2.2.3 Partial derivatives approximation	56
2.2.4 Numerical scheme	58
2.3 Tridiagonal matrix system and its resolution	63
2.3.1 Diagonal coefficients for Neumann boundary conditions	64
2.3.2 Diagonal coefficients for Dirichlet boundary conditions	67
2.4 Algorithm	68
2.4.1 Initial conditions	68
2.4.2 Age initialisation	68

2.4.3	Main loop	68
References		69
3	Configurations and model runs	71
3.1	System requirements	71
3.2	Installations	71
3.3	Running SEAPODYM with predefined configuration	72
3.3.1	Input directories	73
3.3.2	Simulation run	75
3.4	Parameter file	75
3.4.1	Spatial domain and resolution	75
3.4.2	Land mask	76
3.4.3	Time control	77
3.4.4	Forcing data	77
3.4.5	Fisheries configuration	81
3.4.6	Likelihood function configuration	82
3.4.7	Dynamic model parameters	84
3.5	Running modes	94
3.5.1	Simulation	95
3.5.2	Habitats	95
3.5.3	Density	96
3.5.4	Optimization	96
3.5.5	Likelihood profiling	97
3.5.6	Hessian computation	98
3.5.7	Sensitivity analyses	98
3.6	SEAPODYM Outputs	99
3.6.1	Setting the outputs directory	99
3.6.2	Outputs in DYM files	99
3.6.3	ASCII output files	102
References		107
4	Model parametrisation	109
4.1	Integrating data in a fish population dynamics model	110
4.1.1	Effort and catch data	110
4.1.2	Length frequency data	114
4.1.3	Conventional tagging data	115
4.2	Sensitivity of model parameters	121
4.2.1	Local sensitivity analysis	121
4.2.2	Global sensitivity analysis	123
4.3	Maximal likelihood estimation method	126
4.3.1	Adjoint computer code	128

4.3.2	Parameter scaling	130
4.3.3	Likelihood functions in SEAPODYM	130
4.4	Parameter estimation workflow	134
4.4.1	Problem dimensionality and memory control	134
4.4.2	Likelihood configuration	135
4.4.3	Initial conditions	136
4.4.4	Role of environmental forcing	137
4.4.5	Two-dimensional projection of likelihood function	137
4.4.6	Identical twin experiments	138
4.4.7	Correlations between parameters	140
4.5	Error analysis	141
4.6	Model validation	142
	References	144
A	Inventory of model variables and input parameters	147
A.1	Selected notations	147
A.2	XML parfile	149
B	Technical annex	163
B.1	Makefile	163
C	File formats	167
C.1	ASCII fishing data files	167
C.1.1	Effort and catch	167
C.1.2	Size frequencies	168
C.2	ASCII output files	170
C.2.1	SumDym.txt	170
C.2.2	SumQArea.txt	170
C.2.3	SumEEZ.txt	171
C.2.4	MeanVar.txt	172
C.2.5	Spatial_Corr.txt	172
C.2.6	LF_obs.txt	173
C.2.7	LF_Q_fishery.txt	173
C.2.8	LF_Q_sum.txt	174
C.3	DYM files	174
C.3.1	Structure of a DYM file	174
D	Toolbox	179
D.1	SeapodymView	179
D.1.1	Visualization of DYM variables	179
D.1.2	Temporal and spatial extractions	180

D.1.3	Arithmetic operations	182
D.1.4	From DYM to ASCII file	182
D.2	Grid and Mask Builder	182
D.3	R scripts	185

Introduction

The Spatial Ecosystem And POpulation DYnamics Model (SEAPODYM) is being continuously enhanced to provide a general framework allowing integration of biological and ecological knowledge of migratory species, primarily tunas and potentially other oceanic top predator species, within a comprehensive description of the pelagic ecosystem. It includes detailed relationships between the population dynamics and basic biological and ecological functions, a realistic representation of the vertical oceanic habitat in terms of both physical and foraging conditions. The forage fields are predicted by a separate model in which various mid-trophic level organisms (micronekton) are classified by their diel migration pattern, and the spatiotemporal transfer of energy from oceanic primary productivity to the micronekton is described using an allometric scaling equation and passive transport of the biomass with oceanic water masses. The environmental variables that drive fish dynamics (temperature, currents, oxygen and primary production) are predicted by coupled physical–biogeochemical models. The model also includes a rigorous mathematical parameter estimation procedure using available catch and size frequency data. Because the model includes detailed representation of the biophysical environment of the species, the complete spatially explicit population dynamics can be described with a small number of parameters. In return, such a model depends strongly on the quality of environmental forcing variables.

When considering the definition of an ecosystem, that is, how assemblages of species are organised in space and time, and how they interact with each other and the physical environment, modelling the ocean pelagic ecosystem is obviously a challenge that requires drastic simplifications. These simplifications need to be considered carefully alongside the level of observations and knowledge that we have for each component of the system, to make sure that the model can be properly parametrised and can adequately describe observed processes. In addition, the model is focusing on the population dynamics and the fisheries of exploited species, as there is a special interest to provide a new generation of modelling tools for the management of these species, taking into account not only the impact of the fisheries but also the natural fluctuations of the populations in their climate-driven ecosystems. Top predators in the marine pelagic ecosystem are essentially opportunistic omnivorous predators. Their diets reflect both the faunal assemblage of the components of the ecosystem that they explore and their aptitude to capture prey species during different periods of the day

(i.e., daytime, night-time and twilight hours). It seems that most of them are in the upper layer during the night. But high sensory specialisation (e.g., olfaction in sharks, vision in bigeye tuna, swordfish and cephalopods, and echolocation in marine mammals), and morphological and physiological adaptations (e.g., thermoregulation) allow them also to exploit the darker and colder, deeper layers. The three-layer ocean definition used for the mid-trophic (forage) species seems to match particularly well with the known vertical behaviour of large predators (Figure I.1).

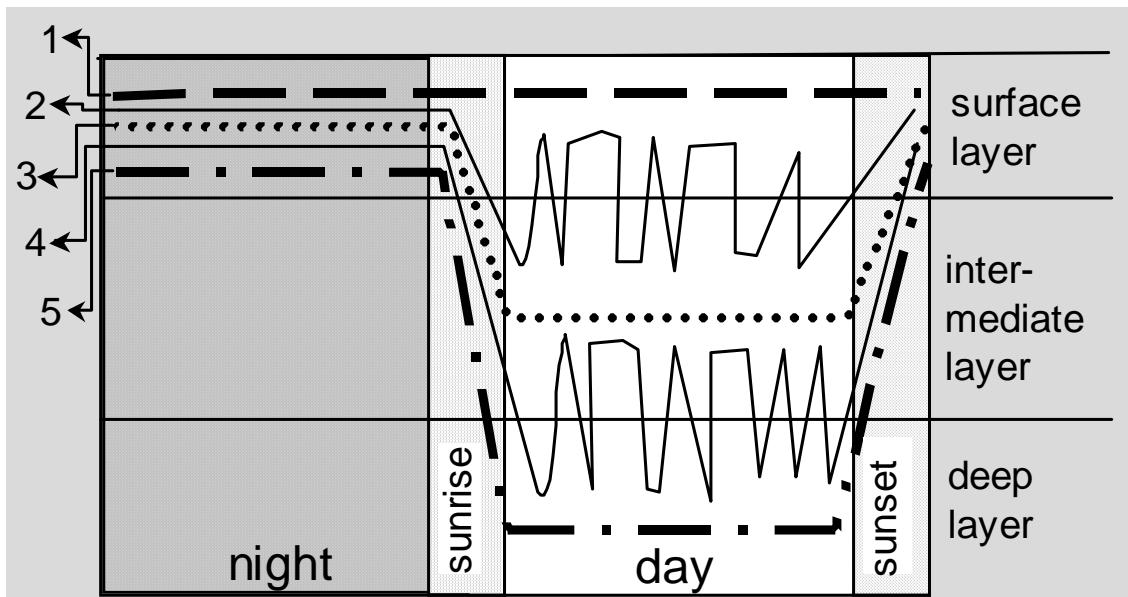


Figure I.1: Five typical vertical movement behaviours simulated using a three-layer and two-type-of-prey pelagic system (adapted from Dagorn et al. 2000): 1, epipelagic predators (e.g., skipjack tuna, marlins and sailfish); 2, predators moving between the surface and intermediate layers during the day (e.g., yellowfin tuna); 3, predators mainly in the intermediate layer during the day (e.g., albacore tuna); 4, predators moving between deep and intermediate layers during the day (e.g., blue shark); 5, predators mainly in the deep layer during the day (e.g., bigeye tuna and swordfish).

Large pelagic predators are often exploited species for which there is detailed knowledge on biology, physiology, population structure and fisheries data. Describing the spatial dy-

namics of these fish populations at oceanic scales is of paramount importance for fisheries management, so as to understand and predict the consequences of fishing, climate change and changes in fishery management regulations. Many spatially explicit models have been developed for fish populations; however, most of them are Lagrangian models considering only passive movements of small fish or early life stage individuals transported by currents (see, e.g., DeAngelis and Yurek, 2016; Rossi et al., 2014; Popova et al., 2019; Ramesh et al., 2019; van Sebille et al., 2018). Parametrisation of Lagrangian models can only be done by manual calibration, using empirically derived parameter values or relying on estimates from external models (as in, e.g., Scutt Phillips et al., 2018). Advection-diffusion-reaction (ADR) equations were suggested by many mathematical ecologists as an elegant solution to deal with complex spatio-temporal patterns arising from animal behaviour (see, e.g., Keller and Segel, 1971; Okubo, 1980; Okubo and Levin, 2001; Murray, 1989; Grunbaum, 1998, 1999; Flierl et al., 1999; Tyutyunov et al., 2004; Tyutyunov and Titova, 2017; Berezovskaya and Karev, 2013; Petrovskii et al., 2002). While ADR equations are unsuitable for low-abundance populations sparsely distributed in space, they provide a very convenient framework for modelling spatial dynamics of migratory species that occupy vast oceanic regions and may create low- and high-density concentrations. The movement dynamics within ADR equations are governed by the distribution of stimuli attracting or repulsing the population density. This classical approach has been thoroughly studied in theoretical movement ecology, widely used in various modelling applications and its interconnections with individual movements are well understood (Okubo et al., 1977; Okubo, 1980; Grunbaum, 1994, 1998, 1999; Flierl et al., 1999; Turchin, 1998; Faugeras and Maury, 2007; Tyutyunov et al., 2013; Tyutyunov and Titova, 2017).

The advantages of using ADR equations can be summarised as follows. While predicting the trajectory of a single individual is impossible due to the stochastic component of its movement, the evolution of density distributions of a large number of individuals can be effectively predicted in a continuous advection–diffusion framework (Grunbaum, 1999; Flierl et al., 1999; Tyutyunov et al., 2004). In addition, explicitly modelling movement not only leads to a reduced number of model parameters and simplified local (in every position in space) functional responses (e.g., Arditi et al., 2001; Tyutyunov and Titova, 2017), but also allows considering simple aggregated statistics (e.g., mean squared step length) in model parametrisations (Grunbaum, 1998). Moreover, incorporating biological-physical interactions by accounting for the impact of temporal and spatial environmental variability on population dynamics and thus reducing the number of model parameters is more straightforward in ADR equations (e.g., Grunbaum, 1998; Flierl et al., 1999) than in classical, spatially aggregated stock assessment models operating in large management regions (Punt, 2017). In exchange for their parsimony, spatially explicit dynamic models applied to full-life-cycle fish population dynamics have higher dimensionality. The latter represents the cost for being more realistic models.

Being continuous, ADR equations allow implementation of quantitative methods to esti-

mate model parameters from available observations (Sibert et al., 1999; Senina et al., 2008). Nevertheless, before applying these models to solve fishery management problems, we need to be confident in the reliability of model predictions. Hence, we need to work on developing the dynamic model, providing adequate descriptions of observed phenomena, and improving the model structure, allowing quantitative dynamics, under the tight coupling of the model predictions to observations.

The SEAPODYM modelling approach for top predator population dynamics has greatly benefited from the work undertaken by Sibert et al. (1999) on implementation of continuous ADR equations to describe movement dynamics and estimate movement and mortality rates of tagged skipjack and yellowfin tunas (see also Sibert and Fournier, 1994). SEAPODYM has been continuously evolving since the initial development of a single-cohort tuna population model presented in Bertignac et al. (1998). A few years later Lehodey (2001) proposed a coupled predator–prey model with the model for tuna forage. In the study of Lehodey et al. (2003), discrete-type age structured equations were added to the numerical solver of continuous ADR equations. The first fully parametrised and fully operational SEAPODYM model was presented in Lehodey et al. (2008), and with an added parameter estimation method in Senina et al. (2008). The general scheme of the model and parameter estimation approach is shown in Figure I.2.

The current model version is based on a continuous ADR equation with an ageing term, and is fully described in Senina et al. (2020a,b). The SEAPODYM modelling framework now includes several models, and allows, besides the full population dynamics model, computing and estimating the parameters of spawning and feeding habitats, modelling movement dynamics of tagged cohorts, and integrating tagging data to inform movement rates of modelled fish at different ages/sizes. The model can be now be parametrised through integration of industrial fisheries data and conventional as well as archival tagging data, and optimization is carried out using the maximum likelihood estimation approach. The implementation of adjoint code allows an exact, analytical evaluation of the likelihood gradient to be obtained. The approach to select the ‘best parameter estimate’ is based on a series of computer experiments in order to i) determine model sensitivity with respect to variable parameters and, hence, investigate their observability; ii) estimate observable parameters and their errors; iii) justify the reliability of found solutions; iv) validate the parametrisation with independent information; and v) compute the errors of parameter estimates.

The continuous improvements in the model structure, the implementation of the numerical model coupled with the quantitative methods of parameter estimation allowed numerous model applications to different pelagic species such as Pacific skipjack tuna (Lehodey et al., 2013; Senina et al., 2020b), Pacific yellowfin tuna (Senina et al., 2015), Pacific bigeye tuna (Senina et al., 2021), Atlantic albacore tuna (Dragon et al., 2015; Senina et al., 2020a), South Pacific albacore tuna (Lehodey et al., 2015; Senina et al., 2020a), Chilean jack mack-

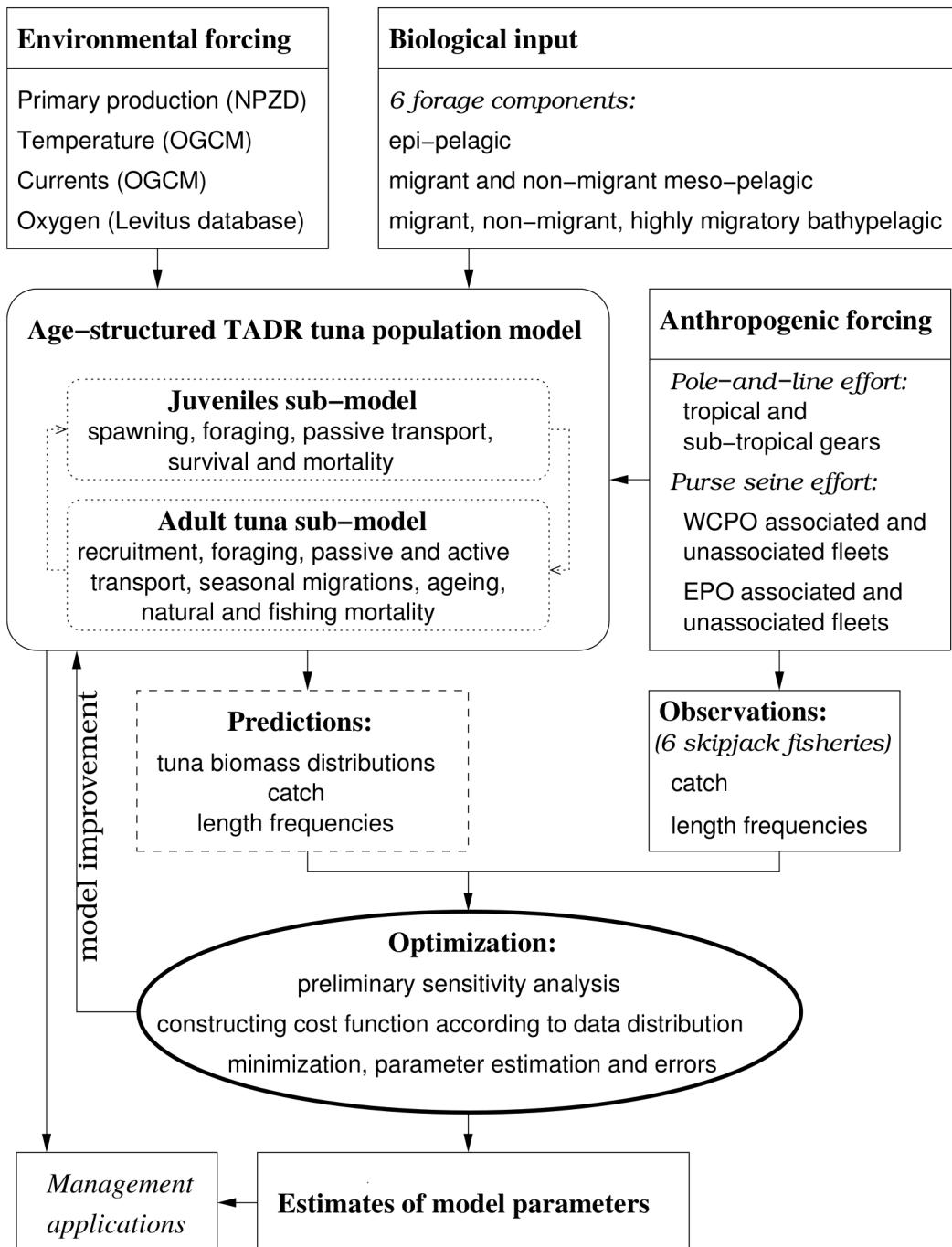


Figure I.2: General scheme of the model with optimization approach (from Senina et al., 2008).

erel (Dragon et al., 2017) and Pacific swordfish (Abecassis et al., 2013). Informing model parameters through integration of various types of data for different migratory species thus showed that such models have the capacity to provide valid quantitative predictions of the species population dynamics, and can be used in the development of management strategies (Sibert et al., 2012) and in investigating the climate change impacts (Lehodey et al., 2010, 2013, 2015; Bell et al., 2021). All examples of applications and related projects can be found in the dedicated websites (www.seapodym.eu; www.spc.int/ofp/seapodym).

This reference manual for the SEAPODYM-MASS model is constructed as follows. Chapter 1, "**The fish population dynamics model**" is devoted to the mathematical model and the dynamic processes it describes. It also details the underlying biological mechanisms behind the definitions of spawning and feeding habitats. Chapter 2, "**Discretisation and numerical solution**" describes the discretisation of model dimensions, the numerical model that is the approximation of the continuous ADR equations with an ageing term, its integration with alternating-direction implicit (ADI) method and the general algorithm implementation within the main time and age loop. Chapter 3, "**Configurations and model runs**" provides detailed documentation on how to run the model simulation with predefined configurations. The input and output data files are also described in this chapter. Chapter 4, "**Model parametrisation**" reviews the methods to build up quantitative models. This final chapter is intended for advanced use of the model, providing the methods for the model application to the new species.

References

- Abecassis, M., Senina, I., Lehodey, P., Gaspar, P., Parker, D., Balazs, G., Polovina, J., 2013. A model of loggerhead sea turtle (*Caretta caretta*) habitat and movement in the oceanic North Pacific. *PLoS One* 8(9), e73274. <https://dx.doi.org/10.1371/journal.pone.0073274>
- Arditi, R., Tyutyunov, Yu., Morgulis, A. Govorukhin, V., Senina, I. 2001. Directed movement of predators and the emergence of density-dependence in predator-prey models. *Theoretical Population Biology*. 59(3): 207-221.
- Bell, J. B., Senina, I., Adams, T., Aumont, O., Calmettes, B., et al. 2021. Pathways to sustaining tuna-dependent Pacific Island economies during climate change. *Nature Sustainability*. <https://doi.org/10.1038/s41893-021-00745-z>
- Berezovskaya, F. S., Karev. 2013. Bifurcation Approach to Analysis of Travelling Waves in Some Taxis–Cross-Diffusion Models. *Mathematical Modelling of Natural Phenomena* 8(3), 133-153. <https://doi.org/10.1051/mmnp/20138309>

- Bertignac, M., Lehodey, P. Hampton, J. 1998. A spatial population dynamics simulation model of tropical tunas using a habitat index based on environmental parameters. *Fish. Oceanogr.* 7: 3/4, 326-334.
- Dagorn L., Bach P., and Josse E. 2000. Movement patterns of large bigeye tuna (*Thunnus obesus*) in the open ocean, determined using ultrasonic telemetry. *Marine Biology* 136: 361-371.
- DeAngelis, D., Yurek, S., 2016. Spatially explicit modeling in ecology: a review. *Ecosystems* 20(2): 284-300. <https://doi.org/10.1007/s10021-016-0066-z>
- Dragon, A.-C., Senina, I., Titaud, O., Calmettes, B., Conchon, A., Arrizabalaga, H., Lehodey, P. 2015. An ecosystem-driven model for spatial dynamics and stock assessment of North Atlantic albacore. *Can. J. Fish. Aquat. Sci.* 72(6): 864–878. <http://dx.doi.org/10.1139/cjfas-2014-0338>
- Dragon, A.-C., Senina, I., Hintzen, N., and Lehodey, P. 2017. A South-Pacific basin-scale modelling of jack mackerel under combined impacts of fishing and climate variability. *Fish. Oceanogr.* 27(2): 97–113.
- Faugeras B. and Maury O. 2007. Modeling fish population movements: from individual-based representation to an advection-diffusion equation. *Journal of Theoretical Biology* 247: 837–848.
- Flierl, G., Grünbaum, D., Levin, S., Olson, D., 1999. From individuals to aggregations: the interplay between behavior and physics. *J. Theor. Biol.* 196(6), 397–454. <https://doi.org/10.1006/jtbi.1998.0842>
- Grünbaum D. 1994. Translating stochastic density-dependent individual behavior with sensory constraints. *Mathematical Models in Biology*. New York: McGrawHill.
- Grünbaum D. 1998. Using spatially explicit models to characterize foraging performance in heterogeneous landscapes. *The American Naturalist* 151(2): 97–115.
- Grünbaum, D. 1999. Advection-diffusion equations for generalized tactic searching behaviors. *Math. Biol.* 38: 169-194.
- Keller, E., Segel, L. A. 1971. Travelling bands of chemotactic bacteria: a theoretical analysis. *J. Theor. Biol.* 30, 235-248.
- Lehodey, P. 2001. The pelagic ecosystem of the tropical Pacific Ocean: dynamic spatial modeling and biological consequences of ENSO. *Progress in Oceanography*. 49, 439-468.

- Lehodey, P., Chai, F., Hampton, J. 2003. Modelling climate-related variability of tuna populations from a coupled ocean biogeochemical-populations dynamics model. *Fish. Oceanogr.* 12: 4(5), 483-494.
- Lehodey P., Senina I. and Murtugudde R. 2008. A spatial ecosystem and populations dynamics model (SEAPODYM) — modeling of tuna and tuna-like populations. *Progress in Oceanography* 78: 304–318. <https://doi.org/10.1016/j.pocean.2008.06.004>
- Lehodey, P., Senina, I., Sibert, J., Bopp, L., Calmettes, B., Hampton, J., Murtugudde, R. 2010. Preliminary forecasts of population trends for Pacific bigeye tuna under the A2 IPCC scenario. *Progress in Oceanography*, Volume 86: 302-315.
- Lehodey, P., Senina, I., Calmettes, B., Hampton, J., Nicol, S. 2013. Modelling the impact of climate change on pacific skipjack tuna population and fisheries. *Climatic Change*, 119(1):95-109. <https://doi.org/10.1007/s10584-012-0595-1>
- Lehodey, P., Senina, I., Nicol, S., Hampton, J., 2015. Modelling the impact of climate change on south pacific albacore tuna. *Deep-Sea Research Part II: Topical Studies in Oceanography* 113:246–259. <https://doi.org/10.1016/j.dsr2.2014.10.028>
- Murray J.D. 1989. Mathematical biology. Berlin: Springer-Verlag. 767 p.
- Okubo A., Chiang H.C., Ebbesmeyer C.C. 1977. Acceleration field of individual midges, Anarete pritchardi (Diptera: Cecidomyiidae), within a swarm. *The Canadian Entomologist* 109:149–156. <https://doi.org/10.4039/Ent109149-1>
- Okubo, A. 1980. Diffusion and ecological problems: mathematical models. Springer-Verlag, NY.
- Okubo, A. and Levin, S.A. 2001. Diffusion and ecological problems: Modern perspectives. New York: Springer. <https://doi.org/10.1007/978-1-4757-4978-6>
- Petrovskii, S., Morozov, A., Venturino, E. 2002. Allee effect makes possible patchy invasion in a predator-prey system. *Ecology Letters* 5 (3), 345-352.
- Popova, E., Vousden, D., Sauer, W., Mohammed, E., Allain, V., Downey-Breedt, N., Fletcher, R., Gjerde, K., Halpin, P., Kelly, S., Obura, D., Pecl, G., Roberts, M., Raitsos, D., Rogers, A., Samoilys, M., Sumaila, U., Tracey, S., Yool, A. 2019. Ecological connectivity between the areas beyond national jurisdiction and coastal waters: Safeguarding interests of coastal communities in developing countries. *Marine Policy* 104: 90-102. <https://doi.org/10.1016/j.marpol.2019.02.050>
- Punt A.E. 2017. Modelling recruitment in a spatial context: a review of current approaches, simulation evaluation of options, and suggestions for best practices. *Fisheries Research* 217:140–155. <https://doi.org/10.1016/j.fishres.2017.08.021>

- Ramesh, N., Rising, J., Oremus, K. 2019. The small world of global marine fisheries: The cross-boundary consequences of larval dispersal., *Science* 364: 1192-1196. <https://doi.org/10.1126/science.aav3409>
- Rossi V., Ser-Giacomi E., Lopez C. and Hernandez-Garcia E. 2014. Hydrodynamic provinces and oceanic connectivity from a transport network help designing marine reserves. *Geophysical Research Letters* 41:2883–2891. <https://doi.org/10.1002/2014GL059540>
- Scutt Phillips J., Sen Gupta A., Senina I., van Sebille E., Lange M. et al. 2018. An individual-based model of skipjack tuna (*Katsuwonus pelamis*) movement in the tropical Pacific Ocean. *Progress in Oceanography* 164:63–74. <https://doi.org/10.1016/j.pocean.2018.04.007>
- Senina, I., Sibert, J., and Lehodey, P. 2008. Parameter estimation for basin-scale ecosystem-linked population models of large pelagic predators: Application to skipjack tuna. *Prog. Oceanogr.* 78: 319–335. <https://doi.org/10.1016/j.pocean.2008.06.003>
- Senina, I., Lehodey, P., Calmettes, B., Nicol, S., Caillot, S., Hampton, J., Williams, P. 2015. SEAPODYM application for yellowfin tuna in the Pacific Ocean. Working paper, 11th Regular Session of the Scientific Committee of the Western Central Pacific Fisheries Commission, Pohnpei, Federated States of Micronesia, 5-13 August 2015. WCPFC-SC11-2015/EB-IP-01.
- Senina, I., Lehodey, P., Hampton, J. and J. Sibert. 2020a. Quantitative modelling of the spatial dynamics of South Pacific and Atlantic albacore tuna populations. *Deep Sea Res. II* 175, <https://doi.org/10.1016/j.dsr2.2019.104667>.
- Senina, I., Lehodey, P., Sibert, J. and J. Hampton. 2020b. Integrating tagging and fisheries data into a spatial population dynamics model to improve its predictive skills. *Can. J. Aquat. Fish. Sci.* 77, 576–593. <https://doi.org/10.1139/cjfas-2018-0470>
- Senina, I., Briand, G., Lehodey, P., Nicol, S., Hampton, J. 2020, Williams, P. 2021. Reference model of bigeye tuna using SEAPODYM with catch, length and conventional tagging data. Information paper, 17th Regular Session of the Scientific Committee of the Western Central Pacific Fisheries Commission, SC17-EB-IP-08. <https://meetings.wcpfc.int/node/12605>
- Sibert, J.R., Hampton, J., Fournier, D.A., Bills, P.J. 1999. An advection-diffusion-reaction model for the estimation of fish movement parameters from tagging data, with application to skipjack tuna (*Katsuwonus pelamis*). *Can. J. Fish. Aquat. Sci.* 56, 925-938.
- Sibert, J.R., Fournier, D.A. 1994. Evaluation of advection-diffusion equations for estimation of movement patterns from tag recapture data. In *Interactions of Pacific tuna fisheries, FAO Summary report and papers on interaction*, 1: 108-121.

- Sibert, J., Senina, I., Lehodey, P., Hampton, J. 2012. Shifting from marine reserves to maritime zoning for conservation of Pacific bigeye tuna (*Thunnus obesus*). *US Proceedings of the National Academy of Sciences*. <https://doi.org/10.1073/pnas.1209468109>
- Turchin, P. 1998. Quantitative analysis of movement. Sinauer, Sunderland, MS.
- Tyutyunov, Yu., Senina, I., Arditi, R. 2004. Clustering due to acceleration in the response to population gradient: a simple self-organization model. *The American Naturalist*. 164(6).
- Tyutyunov Yu., Titova L., Berdnikov S. 2013. A mechanistic model for interference and Allee effect in the predator population. *Biophysics* 58(2):258–264. <https://doi.org/10.1134/S000635091302022X>
- Tyutyunov Yu. and Titova L. 2017. Simple models for studying complex spatiotemporal patterns of animal behavior. *Deep Sea Research Part II: Topical Studies in Oceanography* 140:193–202. <https://doi.org/10.1016/j.dsr2.2016.08.010>
- van Sebille, E. Griffies, S., Abernathey, R., Adams, T. et al. 2018. Lagrangian ocean analysis: Fundamentals and practices, *Ocean Modelling*. 121: 49-75. <https://doi.org/10.1016/j.ocemod.2017.11.008>

Chapter 1

The fish population dynamics model

1.1 Underlying equations

The general model of SEAPODYM is based on a classical advection–diffusion–reaction (ADR) equation. Let $N(a, t, \mathbf{x})$ be the density of the fish population at age $a \in [0, \bar{a}]$, at time $t \in [t_0, t_{\text{fin}}]$ and at position $\mathbf{x} = (x, y) \in \Omega \in \mathbf{R}^2$. Herein for brevity we omit notations of age, time and space, and use the gradient operator $\nabla = (\partial_x, \partial_y)^T$ and the divergence operator of a two-dimensional vector field $\text{div}(\mathbf{v}) = \partial_x u + \partial_y v$. The continuous version of SEAPODYM describing spatial, temporal and age dynamics of fish population density N is represented by the system of ADR equations with an ageing term, and the initial and Neumann boundary conditions:

$$\partial_t N + \partial_a N = -\text{div}(\mathbf{v}N) + \nabla(D\nabla N) - MN + S \quad (1.1)$$

$$N(a, \mathbf{x}, t_0) = N_0(a, \mathbf{x}) \quad (1.2)$$

$$\mathbf{n} \cdot \mathbf{v} \Big|_{\mathbf{x} \in \partial\Omega} = \mathbf{n} \cdot \nabla N \Big|_{\mathbf{x} \in \partial\Omega} = 0 \quad (1.3)$$

where

$\mathbf{v} = \mathbf{v}_c + \mathbf{v}_N$ is the velocity field including velocity of ocean currents \mathbf{v}_c and directed movement velocities of population density \mathbf{v}_N (see eqs. 1.25 and 1.28);

D is the diffusivity (see eq. 1.30);

$M = m_N + m_F$ is the total mortality that is the sum of natural m_N and fishing m_F mortality rates (see eqs. 1.40 and 1.42);

$S = N(0, t, \mathbf{x})$ gives the new recruited biomass after spawning, i.e. at $a = 0$. Note $S = 0$ for $a > 0$ (see eqs. 1.13 and 1.23);

$\partial\Omega$ is the boundary of the two-dimensional domain $\Omega \in \mathbb{R}^2$ and \mathbf{n} is the unit normal vector to $\partial\Omega$;

$N_0(a, \mathbf{x})$ in (eq. 1.2) is the initial value of the state vector, i.e. the population density for all ages $a \in [0, \bar{a}]$ in the two-dimensional space, where \bar{a} is the maximal age of individuals in a given population; and

Neumann zero-flux boundary conditions (eq. 1.3) imply impenetrability of domain boundaries $\partial\Omega$.

In addition to age dynamics, the model (1.1–1.3) describes the dynamics of four life stages. These are larvae, small juveniles, immature (young) and mature (adult) fish (Table 1.1). The reasoning behind considering these life stages is their different movement dynamics. Larvae drift in the surface where they are passively transported by ocean currents; hence their survival depends on the local environmental conditions in the sites to which they are transported. Small juveniles (for tunas, up to 3 months of age, for other fish species, age classes which cannot be considered autonomous in terms of movements) occupy the epipelagic layer, through which they start performing diel migrations to avoid predation. However, their large-scale horizontal movements are still passive and depend on currents in this layer. In the next life stage of young immature fish, in addition to being passively transported by ocean currents, fish can undertake directed movements in search of food. At the fourth life stage, mature adults, the fish have two drivers of movement – survival, that is, searching for food, and reproduction, that is, moving to habitats that provide optimal conditions for spawning and for survival of larvae. Note that dynamics of all life stages are modelled by the general model (eqs. 1.1–1.3); however, the rates of reproduction, mortality and movement are considered differently.

Table 1.1: Considered life stages of a modelled population. The column ‘driver’ specifies the biological driver for active movement (both directed and non-directional).

Life stage	Ages, a	Layer	movement	Driver
Larvae	$a = 0$	surface	passive drift	none
Small juveniles	$0 < a \leq a_J$	epipelagic	passive drift	none
Young adults	$a_J < a \leq a_Y$	all	passive and active	survival
Mature adults	$a_Y < a \leq \bar{a}$	all	passive and active	survival, reproduction

1.2 Dynamic processes

SEAPODYM follows a biophysical approach explicitly describing the spatio-temporal dynamics of a species population density arising from animal behaviour as a response to environment. The environment is described by physical, biogeochemical and biological variables

derived from external models. These variables are temperature, currents, dissolved oxygen, primary production, euphotic depth and density of micronektonic organisms that are the prey of tunas and other large predators (see section 1.2.1). As seen from the model (eqs. 1.1–1.3), the fish population dynamics in SEAPODYM are resolved in four dimensions: two-dimensional space, time and age. The vertical dimension is simplified into three pelagic layers.

Two main drivers of animal behaviour, reproduction and survival, are considered. To integrate them into the model, two types of habitat indices, spawning H_s and feeding H_a , are defined. Definitions of species habitats are based on empirical evidence. A thermal habitat of tuna species is derived from an individual heat budget model. The feeding habitat is computed according to the accessibility of tuna predator cohorts to the different vertically migrating and non-migrating micronekton (mid-trophic) functional groups. The spawning habitat is based on temperature and density of predators and food for larvae in the spawning sites. These habitats, as well as the movements, reproduction and survival, are driven by a biophysical environment predicted from a coupled ocean physical–biogeochemical model.

1.2.1 Species environment

The species environment is described by the spatio-temporal fields of the following variables: i) physical – temperature and ocean currents, ii) biochemical – dissolved oxygen concentration, primary production and euphotic depth, and iii) biological – the food resource for the species (see Table 1.2). The temperature, dissolved oxygen concentration, and zonal/meridional currents are the three-dimensional outputs of coupled ocean general circulation (OGCM) and biogeochemical (BGCH) models. The euphotic depth and integrated primary production are the two-dimensional outputs of either BGCH models or observation-based empirical models. Three pelagic layers represent a simplified vertical dimension of SEAPODYM: the surface epipelagic layer, the subsurface mesopelagic layer and the deep mesopelagic layer. The current definition of these layers is based on the euphotic depth and originates from the definition of six functional groups of micronekton (Figure 1.1).

First, all three-dimensional environmental variables are averaged over three pelagic layers (Figure 1.2). These integrated variables are then used to force the SEAPODYM-LMTL (Lower and Mid-Trophic Level) model. The SEAPODYM-LMTL model relies on primary production, temperature and ocean currents to simulate the biomass of six functional groups of micronekton, that is, mid-trophic-level prey organisms of tunas, residing or migrating through three pelagic layers within the upper 1000 m of the water column. The depth \mathbf{z} of pelagic layers is linked to the depth of euphotic layer z_e as follows:

$$\mathbf{z} = (1.5z_e, 4.5z_e, \min(10z_e, 1000)). \quad (1.4)$$

Table 1.2: Forcing variables used in SEAPODYM applications. The column **Model** refers to the external model type.

Model	Variables	Description
<i>Physical forcing</i>		
OGCM	T, u, v	Ocean reanalysis by general circulation model with atmospheric forcing based on meteorological observations.
<i>Biogeochemical forcing</i>		
BGCH	P, O_2, z_e	Primary production, dissolved oxygen and euphotic depth, either predicted by BGCH model coupled to OGCM, or by empirical model (e.g. VGPM) derived from satellite data.
<i>Biological forcing</i>		
LMTL	F, Z	Six micronekton groups and one lower-trophic level group (zooplankton) predicted by SEAPODYM-LMTL model with the above forcing, excluding dissolved oxygen.

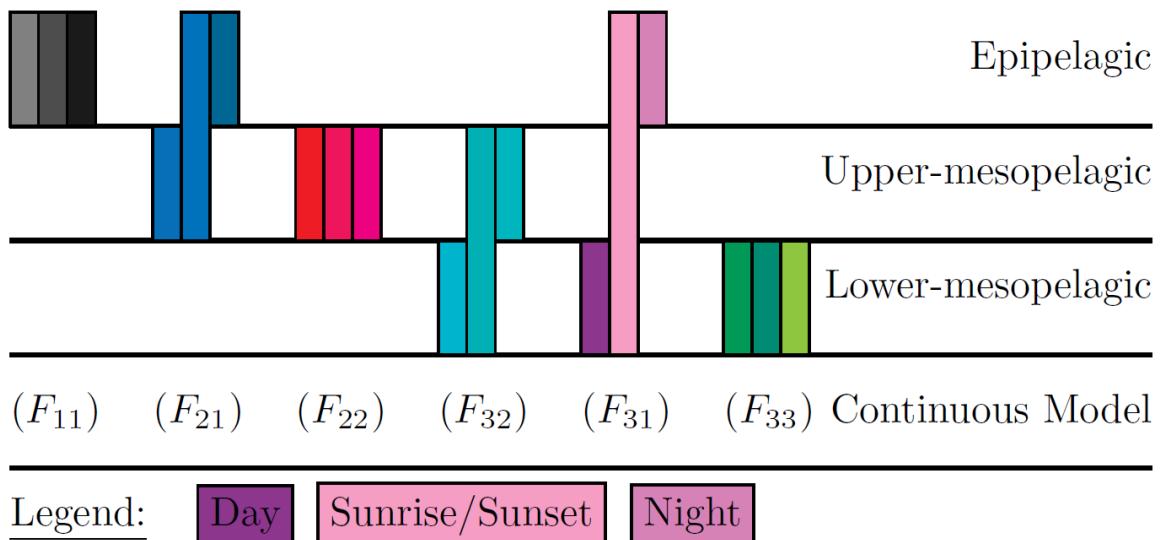


Figure 1.1: The definition of micronekton functional groups by their vertical diel behaviour.

The definition (1.4) of pelagic layers in SEAPODYM is derived from the diurnal patterns in vertical distributions of micronektonic species shown by acoustic observations (see Lehodey et al. (2015) for further details on vertical layer definition and the SEAPODYM-LMTL model). The spatio-temporal fields of modelled micronekton together with the physical and biochemical variables are used to describe the preferred habitats of the large pelagics for foraging and spawning and to predict their temporal and spatial dynamics.

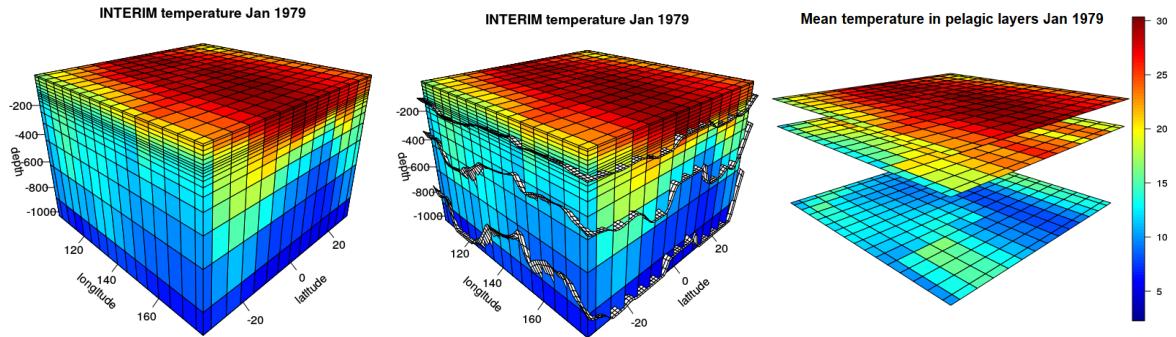


Figure 1.2: The integration of three-dimensional environmental variables. Here three-dimensional temperature (left) is integrated over three pelagic layers (middle) and divided by each layer's thickness to provide the two-dimensional mean fields of each forcing variable.

Let us denote $\text{var}_{\mathbf{z}}(t, \mathbf{x})$, the forcing variable averaged over vertical layer \mathbf{z} , where var can be either water temperature T , dissolved oxygen O_2 , or ocean currents u and v . As mentioned above, the primary production $P = P(t, \mathbf{x})$ is integrated through the whole water column (knowing that most of it is concentrated in the euphotic layer). Vertically integrated primary production is given in units of $\text{mmol C m}^{-2}\text{d}^{-1}$. The density of micronekton $F_{z_d z_n} = F(t, \mathbf{x}, z_d, z_n)$ refers to the modelled density of small nektonic organisms (representing prey of modelled predatory species) aggregated into a functional group by their vertical behaviour, that is, all inhabiting layer z_d and z_n during the day and night respectively, as shown in Figure 1.1. For brevity, hereafter we omit dimensional notations (time, 2D space) for SEAPODYM forcing variables once they are defined. Also, for convenience let us represent all prey functional groups as a diagonal matrix of dimension $\text{dim}(\mathbf{z}) \times \text{dim}(\mathbf{z})$,

$$\mathbf{F} = \begin{pmatrix} F_{11} & 0 & 0 \\ F_{21} & F_{22} & 0 \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \quad (1.5)$$

where the rows and columns show the composition of vertical layers during the day and night respectively.

1.2.2 Spawning habitat

The spawning habitat describes the ensemble of environmental conditions that are favourable for spawning and optimal for larvae survival. In other words, the spawning habitat describes the ensemble of conditions that constrain larval production and mortality, and affect the subsequent recruitment. This habitat then represents the following four mechanisms:

- changes in the spatial extent of the spawning habitat with temperature;

- coincidence of spawning with presence or absence of food for larvae (micro-zooplankton, approximated by primary production), that is, the match/mismatch mechanism proposed by Cushing (1975);
- coincidence of spawning with presence or absence of predators of larvae (that are the micronektonic organisms, i.e., the prey of adults); and
- redistribution of larvae by the oceanic circulation, which can retain larvae in favourable areas with lower natural mortality, or conversely move the larvae to unfavourable zones where the natural mortality will be higher.

The favourability of tuna habitat in terms of spawning success and larvae survival depends on three oceanic variables – sea surface temperature ($SST(t, \mathbf{x})$), prey of larvae ($\Lambda(t, \mathbf{x})$, can be either plankton density or primary production as a proxy), and density of predators of larvae present in the surface layer, which are also the food for spawners ($F_1(t, \mathbf{x})$, surface micronekton, see below).

The index of spawning habitat, $H_s(t, \mathbf{x}) \in (0, 1)$, is defined as the following product (for brevity, we omit here the dimensional notations of environmental variables, micronekton density and habitats):

$$H_s = f_1(SST; T^*, \sigma) \times f_2(\Lambda, \alpha) \times f_3(F_1, \alpha_F, \beta_F), \quad (1.6)$$

requiring that all three environmental conditions are optimal for the species for the habitat index to be maximal. These three functions (see Figure 1.3) are defined between 0 and 1 as follows. The thermal conditions are described by

$$f_1 = e^{-\frac{(SST - T^*)^2}{2\sigma^2}}, \quad (1.7)$$

a Gaussian function with two parameters, T^* and σ , being the optimal temperature and thermal tolerance interval of larvae respectively. The function f_2 is the analogue of the Holling type III functional response function with $n = 2$:

$$f_2 = \frac{\Lambda^n}{\alpha + h\Lambda^n}, \quad (1.8)$$

where handling time $h = 1$ (to allow the function scaling within $[0, 1]$ interval) and $\alpha \geq 0$ the inverse of searching efficiency defining the shape of functional response to the prey densities (no response with $\alpha = 0$, mostly hyperbolic within the effective range of prey densities as the inflection point $\Lambda = \sqrt{\frac{\alpha}{3}}$ gets too close to 0 with small α and marked logistic relationship with large enough α). Note that when using primary production P as a proxy for the density of phyto- and zooplankton, which are the actual prey of larvae, we should convert primary production to the wet weight of plankton. So if ocean primary production P is given in

$\text{mmol}\cdot\text{C}\cdot\text{m}^{-2}$, we use the energy transfer constant $E = 0.354$ to compute the part of P that will be transferred to lower-trophic level groups (see Lehodey et al., 2015) and the conversion factor $c = 0.1415 \text{ gWW}\cdot\text{mmol}^{-1} \text{ C}$ (Iverson, 1990), so that $\Lambda = cEPt$, the t being the unit time of production P , provides the wet weight of plankton species in the units of $\text{g}\cdot\text{m}^{-2}$.

The third function, f_3 , is the log-normal distribution function rescaled on $(0, 1)$ allowing selection of the optimal window of micronekton densities F_1 in the surface layer:

$$f_3 = \frac{1}{F_1 e^{0.5\beta_F^2 - \alpha_F}} e^{-\frac{(\log F_1 - \alpha_F)^2}{2\beta_F^2}} \quad (1.9)$$

where parameters α_F and β_F are the mean and standard deviation of log-normal distribution function, and the surface micronekton densities F_1 (1 is the layer index) is computed as follows:

$$F_1 = \tau \mathbf{F}_{1.} \delta + (1 - \tau) \mathbf{F}_{.1}^\top \delta \quad (1.10)$$

with $\delta = (1, 1, 1)^\top$ and τ is the proportion of light hours in a 24-hour cycle. In other words, F_1 is the density of all micronektonic organisms present in the epipelagic layer during the day, $\mathbf{F}_{1.}$, and during the night, $\mathbf{F}_{.1}^\top$, either residing or migrating here from deeper layers. However, larvae predation is likely maximal during daytime and twilight periods, and therefore only two twilight hours are considered in the second term; hence the formula 1.10 simplifies to

$$F_1 = \tau \mathbf{F}_{1.} \delta + \mathbf{F}_{.1}^\top \delta / 12 \quad (1.11)$$

1.2.3 Reproduction

Successful larval recruitment is linked to both the spawning stock density and the conditions of larvae survival. Hence, the number of recruited larvae $N(0, t, \mathbf{x})$ is the product of H_s (eq. 1.6) and the stock-recruitment function of density of mature adults.

The total amount of mature adults in each position of two-dimensional space, $\hat{N}(t, \mathbf{x})$, is computed using the continuous maturity function $\mu = \mu(a)$ obtained from external studies (see section 1.2.4) and the density of adult tuna predicted by the model (for brevity, hereafter we leave notation of age dimension and deliberately omit (time, 2D space) notations for SEAPODYM state variables):

$$\hat{N} = \int_{a_J}^{\bar{a}} \mu(a) N(a) da \quad (1.12)$$

where a_J is the last age in a small juvenile stage and \bar{a} is the maximal age of individuals in the population. The number of new recruits at zero age, $N(0, t, \mathbf{x})$, that is, the term S in

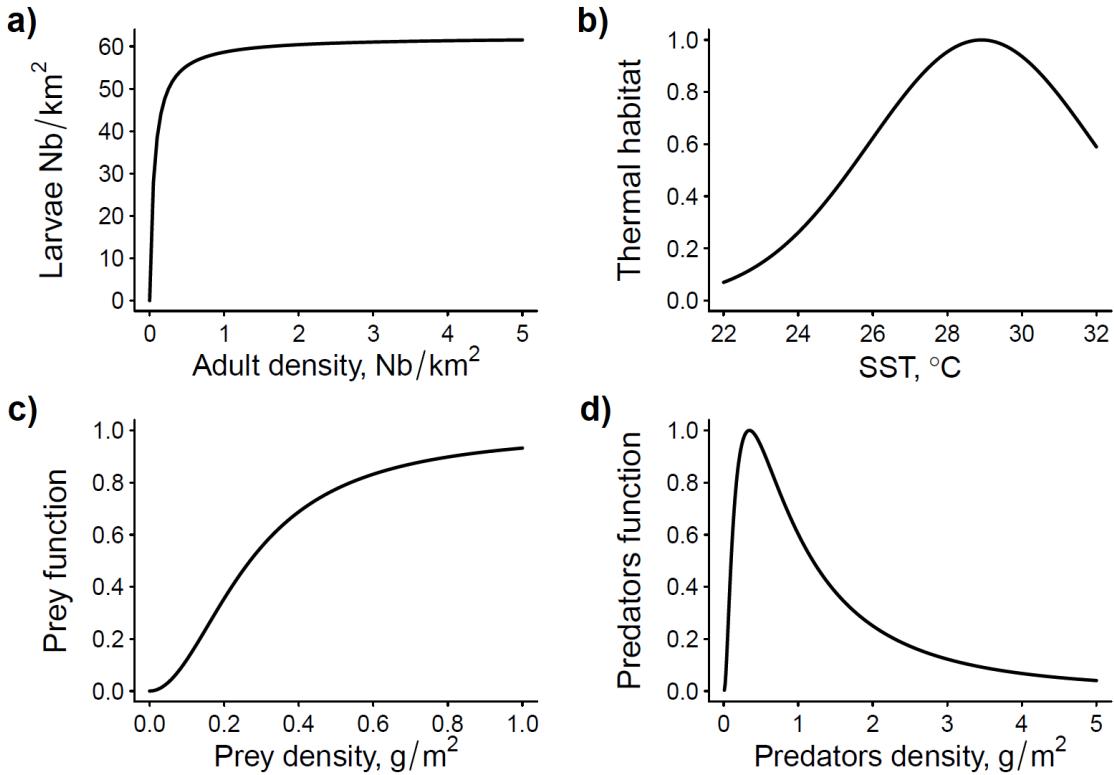


Figure 1.3: Functions used in the definition of spawning habitat and reproduction: (a) Beverton–Holt stock–recruitment function, (b) thermal function f_1 , (c) prey function f_2 , and (d) predator function f_3 , based on parameters estimated for bigeye tuna (Senina et al., 2020b).

eq. 1.1, is then the product of the Beverton–Holt stock-recruitment function (Beverton and Holt, 1957) and the spawning habitat index:

$$N(0) = H_s \frac{r \hat{N}}{1 + b \hat{N}} \quad (1.13)$$

where r is the reproduction rate and b is a parameter defining the strength of the stock–recruitment relationship between the density of spawners, \hat{N} (in $\text{Nb} \cdot \text{km}^{-2}$, Nb - number of individuals), and those of larvae, $N(0)$, survived and recruited to the first age class (Figure 1.3). This relationship is suitable for opportunistic spawners, for which the spawning success depends on the local spawning habitat index and the adult biomass.

1.2.4 Growth and maturity

SEAPODYM uses the species' biological parameters estimated in external studies, which are usually obtained from fitting the classical Bertalanffy (1938) growth equation either to data derived either from conventional tagging or from otolith increments. This equation assumes that as the organism ages, its length increases to an asymptote ℓ_∞ , which is the greatest possible length that the organism can attain under the given conditions:

$$\ell(a) = \ell_\infty \left(1 - e^{(-k(a-a_0))}\right) \quad (1.14)$$

where $\ell(a)$ is length of a species at age a , constant ℓ_∞ is asymptotic length corresponding to the known maximum length of the oldest fish, k is growth rate, and a_0 is age 'zero' length. In SEAPODYM, the relationships length-at-age and its inverse are used in the definitions of habitats, the integration of length frequency and tagging data.

The weight of organisms relates to length-at-age following the general allometric relationship:

$$w(a) = q\ell(a)^p \quad (1.15)$$

with the power constant p estimated by fisheries biologists from weight and length measurements. For many fish and tuna species, the constant $p \approx 3$, meaning that the coefficient q is the fraction of the volume of a cube of side ℓ that the organism occupies (see Beverton and Holt, 1957). The species' weight-at-age function has the asymmetrical sigmoid form and is used in SEAPODYM to compute the species' thermal inertia (section 1.2.5) and in predicting catch statistics (section 4.1.1).

Maturity-at-age is another external (fixed) parameter in SEAPODYM. Two ways of defining maturity in the model are proposed:

- (i) 'knife-edge' maturity, occurring from immature to the mature state at age when 50% of individuals are mature, a_{50} , so that only adults older than age a_{50} (equivalent to a_{Y+1} in Table 1.1) are considered capable of reproducing. This means that in the equation 1.12 of spawning stock biomass, maturity $\mu(a) = 0$ for all $a < a_{50}$ and $\mu(a) = 1$, otherwise;
- (ii) the continuous maturity function $\mu(a)$, giving the estimation of maturity at age expressed in values between 0 and 1 equal to the portion of mature adults at a given age a . In this case the continuous function is used in eq. 1.12.

1.2.5 Feeding habitat and movement rates

The quality of feeding habitat controls the directed component of tuna density movement. We define the **index of feeding habitat as the micronekton density that is accessible to the predator**. This index is therefore not null when the predators and prey share the same habitats and the environmental conditions in the prey's habitat can be tolerated by the predator.

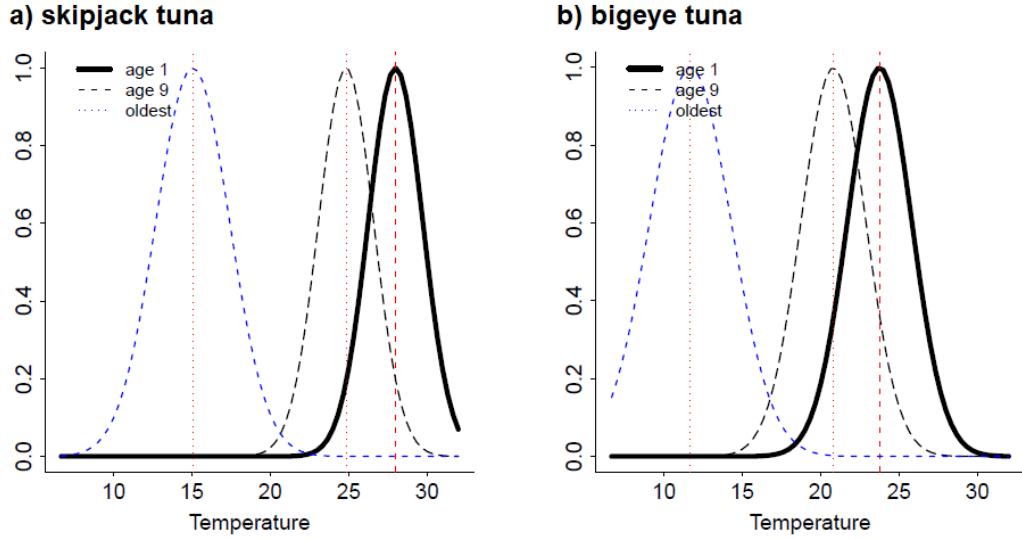


Figure 1.4: Changes in thermal preferences function f_4 linked to species' growth: the optimal temperature decreases with increased body size, while standard deviation increases with increased body weight. Two panels depict examples of typical parametrisation for (a) skipjack and (b) bigeye tuna.

Accessibility

The accessibility of a species to a pelagic layer depends on environmental conditions and the species' preferences and/or tolerances related to these conditions, which vary with age (size). In SEAPODYM, we consider species' thermal preferences and oxygen demands as two main factors stipulating species' accessibility to a given depth and position in two-dimensional space, $\Theta(a, t, \mathbf{x}, z)$. For brevity, hereafter we leave notation of age dimension and deliberately omit (time, space) notations for SEAPODYM variables depending on environment once they are defined. Thus, accessibility $\Theta(a)$ to a given vertical layer is the product of the two functions – temperature and oxygen in the corresponding layer:

$$\Theta(a) = f_4(T; T^*(a), \sigma(a)) \times f_5(O_2; O_2^*, \gamma) \quad (1.16)$$

where f_4 is the Gaussian function selecting the interval of preferred temperatures around the optimal value $T^*(a)$ and the width of the interval depending on $\sigma(a)$ (Figure 1.4):

$$f_4 = e^{\left(-\frac{(T - T^*(a))^2}{2\sigma^2(a)} \right)} \quad (1.17)$$

It is known that species have different tolerances to oxygen levels. The lowering of oxygen concentration can quickly make the habitat unfavourable for tunas, expressed in behavioural

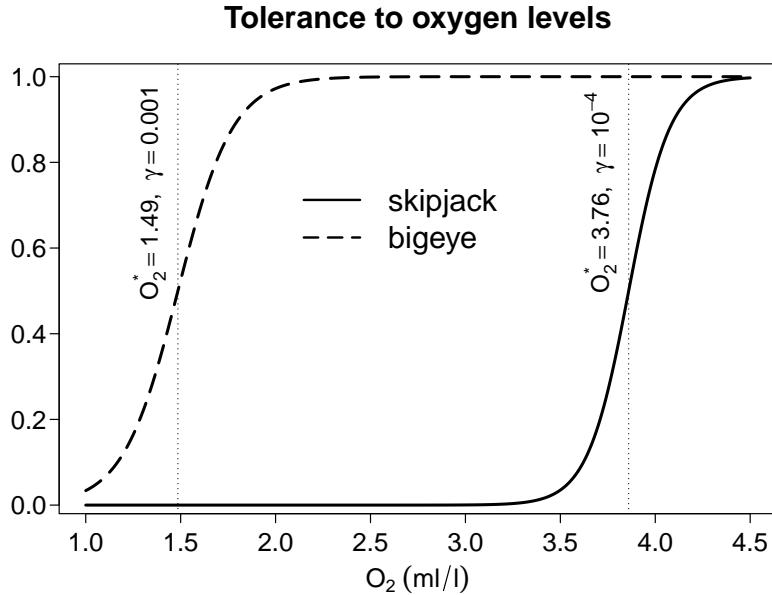


Figure 1.5: Dependence of feeding habitat on oxygen level. Parameters for skipjack and bigeye tunas were obtained with help of MLE method using catch, catch at size and tagging data in the likelihood and objectively analysed oxygen climatologies from World Ocean Atlas (Senina et al., 2018, 2021).

responses such as increased speeds (see, e.g. Brill, 1994). The effect of oxygen on the habitat quality can be described by a sigmoid function, f_5 , stipulating the minimal level of dissolved oxygen necessary for the modelled species (Figure 1.5), with the response to near-critical values of oxygen defined by the parameter γ :

$$f_5 = \frac{1}{1 + \gamma^{(O_2 - O_2^*)}} \quad (1.18)$$

Age dependence

The use of a Gaussian distribution with a preferred temperature mean linked to the fish size-dependent body temperature at steady state, and with a standard error of the distribution linked to the thermal inertia of the fish was formulated in Lehodey et al. (2003) based on earlier works on heat budget models (Holland et al., 1992; Brill, 1994; Maury, 2005). It is assumed that for a given species there is an *optimal intrinsic temperature* (T^*) that remains constant whatever the age/size, and that this temperature is a target temperature for any individual of the species (e.g., due to genetic and physiological adaptation during species evolution). As demonstrated by a tuna heat budget model (Holland et al., 1992), when

becoming larger with age, the fish will have to search for a colder habitat to compensate for their increasing body temperature at steady state. But they will also have a larger temperature range due to their thermal inertia increasing with size $\ell(a)$. Therefore we define the population **size-dependent thermal preference** with optimal temperature decreasing with size and tolerance interval linearly increasing with weight (Figures 1.4, 1.6).

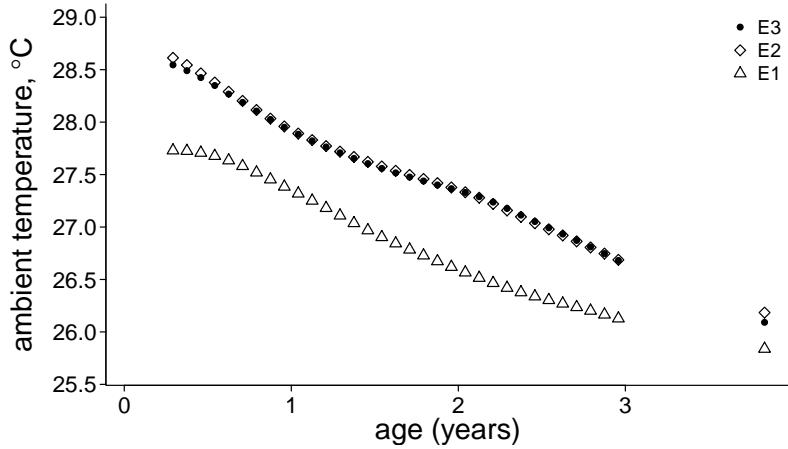


Figure 1.6: Habitat temperature computed as a weighted spatio-temporal mean, with weights being the population density $N(a, t, \mathbf{x})$. Temperature profiles were obtained by means of parameter estimation for skipjack tuna using different datasets and experimental configurations (see Senina et al., 2016, for more details).

Let each age a have corresponding fork-length $\ell(a)$ (eq. 1.14) and weight $w(a)$ (eq. 1.15). Thermal preference implies that the maximal average temperature occurs at age 0, that is, spawning (Figure 1.6). We assume that the species intrinsic temperature $T^* = T_0$ is equivalent to the temperature preferred by the individuals at $a = 0$. Then that thermal preference component of the accessibility function has parameters that are functions of age (through dependence on size) as follows:

$$T^*(a) = T_0 - (T_0 - T_m) \left(\frac{\ell(a)}{\ell(A^+)} \right)^{b_T}, \quad (1.19)$$

$$\sigma(a) = \sigma_0 + (\sigma_m - \sigma_0) \frac{w(a)}{w(A^+)}, \quad (1.20)$$

where T_0 , T_m are the model parameters, denoting temperatures preferred by fish at age 0 and at maximal age (A^+), and σ_0 , σ_m are standard deviations in Gaussian thermal functions (eq. 1.17) for youngest and oldest cohort respectively. The parameter $b \in (1, p)$ allows the

linear to approximately cubic ($p \approx 3$ in eq. 1.15) relationship between preferred temperature and fish size, with $b = 1$ equivalent to a linear decrease with species length at age, $\ell(a)$, and with $b = p$ equivalent to a linear decrease with species weight at age, $w(a)$. Note that, according to the heat budget model outcomes, we may assume that $\sigma_0 = \sigma$ and $T_0 = T^*$ in the spawning habitat definition (eq. 1.7), that is, that the species' preferred range and the optimal intrinsic temperature are the same as the range and the optimal temperature at larval stage. However, due to the use of integrated temperature fields (Fig 1.2), the estimated optimal temperature for spawning might not correspond to the actual temperature at age '0' that the fish are seeking. For this reason, these parameters may be considered separately.

Feeding habitat index

The index of feeding habitat $H_a \in (0, 1)$ is a linear combination of the abundance of the six functional groups of micronekton inhabiting three pelagic layers (eq. 1.5), with the coefficients being the accessibilities to the corresponding pelagic layer (see Table 3.7). However, the biomass levels of micronekton should be considered with caution due to uncertainty associated with the calibration of energy transfer coefficients, which control the total abundance of each functional group (Delpech et al., 2020), and the lack of information to derive the vertical distribution of the biomass from acoustic data (Lehodey et al., 2015). Therefore, one parameter, E_{ij} , per functional group has been added to account for an uncertainty on the absolute value of micronekton biomass density as well as to include the possible effect of food preference of the predator species. Then the matrix \mathbf{F} can be rewritten as:

$$\mathbf{F} = \begin{pmatrix} E_{11} \cdot F_{11} & 0 & 0 \\ E_{21} \cdot F_{21} & E_{22} \cdot F_{22} & 0 \\ E_{31} \cdot F_{31} & E_{32} \cdot F_{32} & E_{33} \cdot F_{33} \end{pmatrix}. \quad (1.21)$$

Finally, knowing the accessibility of the predator at age a to each vertical layer z with prey species, $\Theta(a, z)$, which is the row vector of size 3 (number of vertical layers) at each t and position \mathbf{x} , the feeding habitat at age, at each spatial position and model time step can be written as follows:

$$H_a(a) = \Theta(a) \times (\tau \mathbf{F} \delta + (1 - \tau) \mathbf{F}^\top \delta), \quad (1.22)$$

where \mathbf{F} is the matrix (1.21), $\delta = (1, 1, 1)^\top$, and the weights τ denote the duration (portion) of daytime to account only for the micronekton biomass when it is present in the layer accessible by the predator. Note that the vector product in eq 1.22, $H_a(a, t, \mathbf{x})$, is a scalar field of a unitless quantity. The coefficients E , besides modifying the vertical structure of micronekton groups (although preserving their original two-dimensional distributions predicted by the SEAODYM-LMTL model), have an important physical meaning, that is the inverse of the maximal (asymptotic) value of the density of a given micronekton group, in order to consider the feeding habitat attractive to the predator. The latter theoretically provides that each

term of the linear combination (1.22) is limited within the (0, 1) interval. Besides, due to the Gaussian form of the thermal accessibility function and the definition of vertical layers, two layers cannot be favourable to a species at a given age and point in space and time. So, in practice the habitat index is either the sum of nearly zero contributions of micronekton groups if the conditions in neither layer are favourable, or the selection of micronekton biomass (scaled) in the preferred layer. To ensure in practice that the condition on the resulting index $H_a \leq 1$ is met everywhere and for all age classes, additional scaling using a continuous rotated hyperbola function is applied throughout the model dimensions to reset to 1 all $H_a > 1$ values obtained from the formula (1.22). Since the calibration of parameters E is done by means of constrained function minimisation (Chapter 4), the final habitat is nudged to vary between 0 and 1, and the habitat gradient to provide local cues for predator movements (see section **Active directed movement** below).

Passive movement

We denote $\mathbf{v}_z(t, \mathbf{x})$ the mean horizontal ocean current velocity at time t and position \mathbf{x} , averaged over vertical layer z . The model simulates passive transport of tuna larvae and small juveniles. Densities of these life stages are transported by oceanic currents in the surface (epipelagic) layer, so the total velocity in 1.1, $\mathbf{v} = \mathbf{v}_0$ for ages $[0, a_J]$. Note that for most tuna species the movement dynamics at early life stages is changing rapidly with fish growth. Thus, tuna larvae occupy the upper 50 m of water column (Llopiz et al., 2010), while small juveniles are capable of diel migrations through the euphotic layer. Hence it is more realistic to use the near-surface currents to describe the transport of fish at the larval stage and the currents integrated over the epipelagic layer to model advection of small juveniles. More accurate representation of vertical distribution of these early life stages will allow a more realistic description of their horizontal drift.

Whatever the size of the fish, the horizontal oceanic currents influence the movement of fish living in the water column. To incorporate the drift that the fish may experience while performing vertical diel migrations through layers z , we calculated the integrated current velocity through the water column, accounting for the time fish spend in each layer during the day and night. This time can be assessed as relative feeding habitat index, that is, for each t and \mathbf{x} we first compute the habitat component in each layer as:

$$H_a(a, z) = \Theta(a, z) (\tau \mathbf{F}_{z.} \delta + (1 - \tau) \mathbf{F}_{.z} \delta) \quad (1.23)$$

where \mathbf{F} is matrix (1.21). Then the time spent in each layer can be approximated by the relative contribution of the accessible micronekton density to the total accessible micronekton:

$$\vartheta(a, z) = \frac{H_a(a, z)}{\epsilon + \sum_{z=0}^{z=2} H_a(a, z)} \quad (1.24)$$

where the small constant ϵ is added to avoid division by zero. Then the averaged velocity, weighted by the time spent in the layer is computed as:

$$\tilde{\mathbf{v}}_c(a) = \sum_{z=0}^{z=2} \vartheta(a, z) \mathbf{v}_z, \quad (1.25)$$

with $\vartheta(a)$ and the total velocity in (1.1) is

$$\mathbf{v}_c(a) = \begin{cases} \mathbf{v}_0 & \text{if } a \leq a_J \text{ larvae and juveniles} \\ \tilde{\mathbf{v}}_c(a) & \text{if } a > a_J \text{ adults} \end{cases} \quad (1.26)$$

Active directed movement

The movement dynamics in SEAPODYM are governed by the advection–diffusion terms in model (1.1). Thus, directed movements are modelled with the advection component of the model equation, while random (non-directional) dispersal is described by diffusion. Similar to Lagrangian models of individual behaviour, the movements of population density in a Eulerian approach are stipulated by local conditions only. Repeated responses to local cues at the individual level result in movements towards more favourable parts of the environment. These processes are formulated within Eulerian models by allowing the velocity field of population density to be proportional to the gradient of the movement stimuli. Hence, the velocity of directed movements of fish density, \mathbf{v} , is proportional to the gradient of the vector field of movement stimuli \mathbf{I} :

$$\mathbf{v} = \chi \nabla \mathbf{I} \quad (1.27)$$

with the coefficient of proportionality, the so-called taxis coefficient, χ .

For any type of movements, the displacement per unit of time is directly dependent on the size of the individuals. Since the SEAPODYM-MASS model resolves population dynamics in age, we can easily formulate the movement rates at age. The movement stimuli are the feeding resource's densities becoming increasingly accessible to fish as they grow, which is described by the feeding habitat index (1.22). However, it is also known that the maximal sustainable speed of an individual is linked to its size through physiological mechanisms and not only because of variable environment (see e.g., Cayre, 1991; Brill, 1994; Nihira, 1996; Turchin, 1998). To account for this we introduce a link between the density velocity field and the maximal sustainable speed at age (size) by making the coefficient of proportionality in the classical approach (1.27) a function of age, $\chi(a)$. So the directed component of density velocity field at time t and in position \mathbf{x} is proportional to the gradient of feeding habitat as follows:

$$\mathbf{v}_N(a) = \chi(a) \left(\frac{\partial H_a}{\partial x}, \frac{\partial H_a}{\partial y} \right)^T, \quad (1.28)$$

where $\chi(a) \propto V_{max,a}$, with the maximal sustainable speed, $V_{max,a}$, being the maximal speed of a population density, hence the mean speed of a large number of individuals. It is linked to the mean individual size at age through the allometric relationship $V_{max,a} = V\ell_a^A$, with parameters V expressed in body length per second (BL/s) and the power constant $A < 1$ allowing a decrease of the speeds in $BL \cdot s^{-1}$ with increasing size. Note, to ensure that the resulting speeds from (1.28) do not exceed the $V_{max,a}$, it is necessary that the norm of the velocity $\|\mathbf{v}_N(a)\| = V_{max,a}$ at highest habitat gradients. Thus, denoting $g_m = \max(\|\nabla H_a\|)$,¹ we may formulate the taxis coefficient as

$$\chi(a) = g_m^{-1} V \ell_a^A \quad (1.29)$$

with parameters V and A the control parameters to be estimated with the help of the MLE approach (see Figure 1.7), for example, informing them from movement data, such as conventional tagging (more details about the method in Chapter **Model parametrisation**, section 4.1.3).

Active random movement

The diffusion term in model (1.1) describes random non-directional movements of a large number of individuals. Such movements are made by individuals when there is no preferred direction for movement, or when the local conditions became unfavourable (e.g., no more food). Besides, we can expect that individuals stay longer in the presence of favourable conditions (hence low diffusion), but will want to escape quickly from unfavourable habitats (high diffusion). Diffusion rate, or diffusivity, D in (1.1), is therefore also linked to the feeding habitat index H_a , allowing an increase of non-directional movement velocities in a very poor habitat and minimising random displacements within a favourable habitat. Such a link is formulated as follows for each age a , at time t and position \mathbf{x} :

$$D(a) = \sigma D_0 (1 - cH_a^p(a)) \quad (1.30)$$

where σ is a constant parameter, the coefficient D_0 is the mean theoretical diffusion rate in an unfavourable (zero) habitat, parameter c controls the reduction of D_0 with increasing habitat index; the constant $p = 3$ is chosen to limit the reduction of D_0 in the lowest habitat indices $H_a < 0.5$, making diffusivity less sensitive (always highest) in the poor quality habitat and rapidly increasing in the favourable habitat.

The coefficient D_0 is derived from the theoretical formula for two-dimensional mean square displacement $r^2 = 4Dt$. Thus, if we make the assumption, although unrealistic, that at zero habitat all individuals move at their maximal speed \bar{V} without turning, that is,

¹Note that in the current version of code a simplified formula is implemented for a special case when $\Delta x = \Delta y$.

mean square displacement at the time interval ΔT can be expressed as $r^2 = \frac{1}{n} \sum_{i=1}^n (V_i \Delta T)^2 = \bar{V}^2 \Delta T^2$, then we may compute the maximal theoretical diffusivity as

$$D_0 = \frac{\bar{V}^2 \Delta T}{4}, \quad (1.31)$$

where the average speed of fish is assumed to be $\bar{V} = 1 \text{ } BL \cdot s^{-1}$ for any a . Since this formula obviously overestimates the diffusivity due to the unrealistic assumption above, we add the proportionality constant σ , the meaning of which is the reduction of the maximal theoretical diffusivity (1.31). Note that (1.31) models the non-linear increase of diffusivity with age (see Figure 1.7).

Seasonal spawning migrations

One of the hypotheses of how the search of spawning grounds occurs assumes that adult tuna tend to direct their movements to find a place with environmental conditions like those occurring during their birth (see e.g., Cury, 1994). Another hypothesis, which is used in SEAPODYM modelling, implies that the change in daylight triggers the switch from foraging to spawning behaviour (Lehodey et al., 2008). These two hypotheses can be implemented as follows. At every time t and latitude y , a continuous seasonal switch function $\varsigma(t, y)$ enables rapid switching between two habitats depending on the function of daylight, so that the habitat of mature adults can be computed as a linear combination of feeding and spawning habitats:

$$H = \varsigma H_a + (1 - \varsigma) H_s \quad (1.32)$$

where the switch $\varsigma(t, y)$ is a step function defined as follows:

$$\varsigma(t, y) = (1 + \alpha^{\varrho - \varrho_s})^{-1} \quad (1.33)$$

with constant α defining the abruptness of the switch, $\varrho(t, y)$ is a function of a day length and ϱ_s is the critical value that triggers the switch.

However, the use of the switch function (1.33) and the linear combination of feeding and spawning habitats (1.32) may lead to two problems. First, the predefined seasonality of a daylight function does not necessarily align with the timing of albacore migrations to the spawning grounds. Second, when feeding and spawning habitats are characterised by very different oceanographic conditions, which is often the case for the temperate species (e.g. albacore, bluefin tuna), these two habitats, spawning and feeding, do not overlap geographically and hence there are no local cues that may stimulate long-distance migrations to spawning grounds.

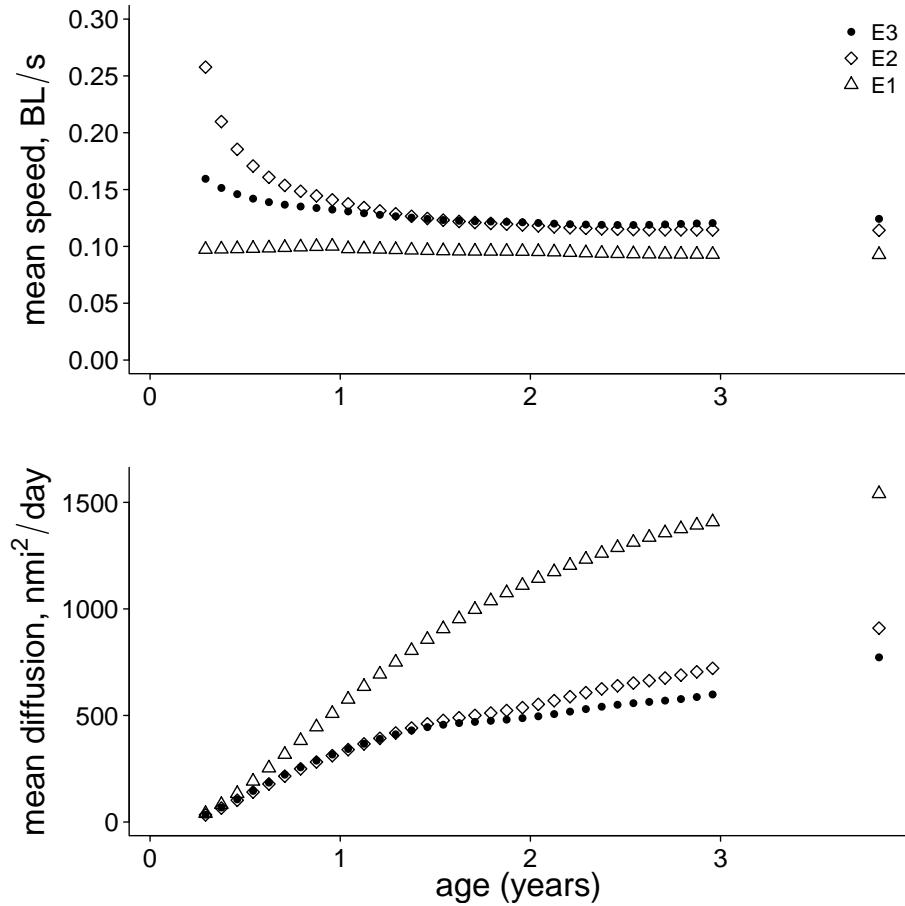


Figure 1.7: Mean movement rates estimated for skipjack tuna using different datasets and experimental configurations: (top) directed velocity of population density in body length per second units at age; (bottom) average diffusion rates at age. The movement rates were averaged over space and time, with weights being spatio-temporal distributions of density $N(a, t, \mathbf{x})$ (see Senina et al., 2016, for more details).

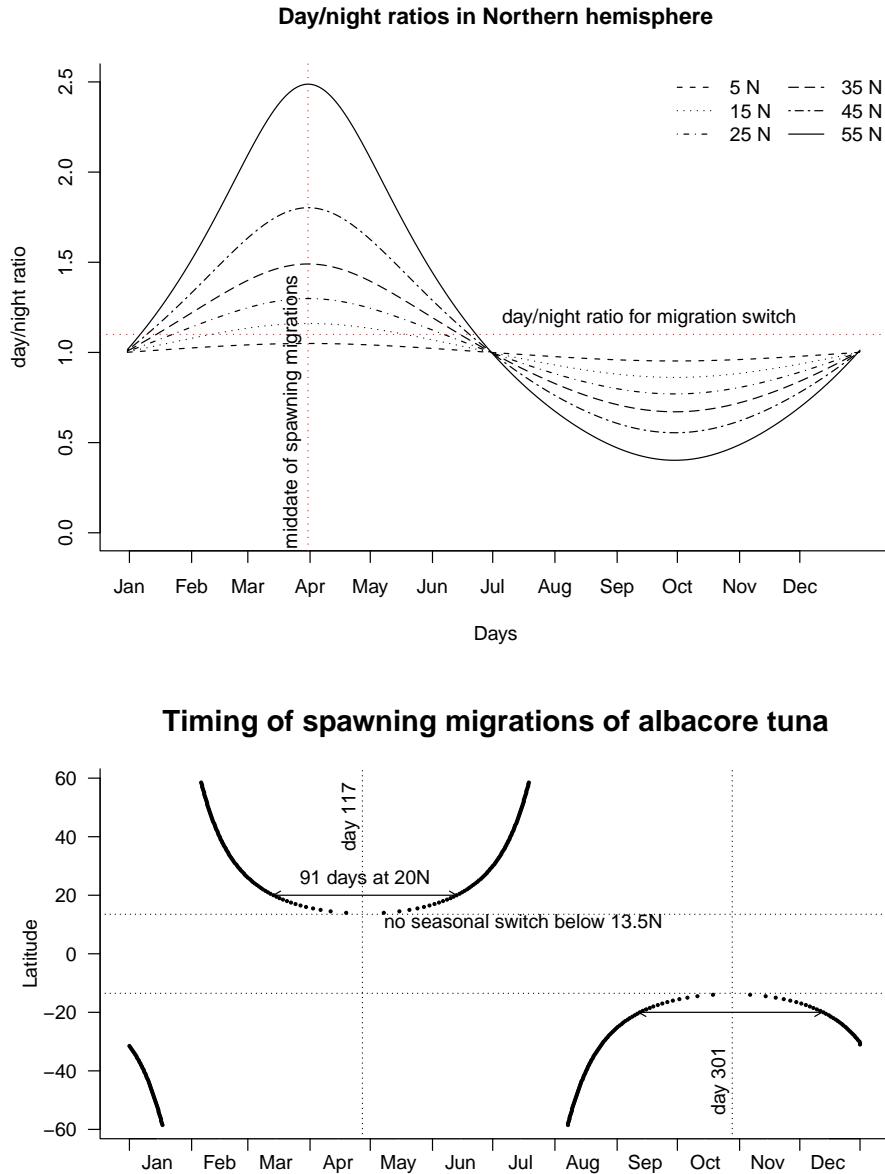


Figure 1.8: Illustration of seasonal switch functions: (top) day-night ratio ϱ for northern hemisphere; (bottom) the dates of seasonal switch eq. 1.33 between feeding and spawning migrations at different latitudes. The dotted vertical lines denote the mid-date, \hat{t} , of the spawning season. The parameter ϱ_s defines the latitude, below which the seasonal switch does not occur.

The flexible timing of the spawning season is enabled by adding a time lag to the day $t\odot$ of summer solstice:

$$\delta t = t\odot - \hat{t}, \quad (1.34)$$

and computing the day length at day $t + \delta t$. So, for the day length $\lambda(t + \delta t, y)$, the \hat{t} becomes the mid-date of the spawning season, also called a season peak. In this case the day length is used only as a convenient periodic function providing the seasonality at every latitude. Note that the choice of mathematical form of the function of day length, ϱ , should be done within the MLE approach given that parameters defining the moment of the switch ϱ_s , and the mid-date of the spawning season \hat{t} can be effectively estimated from the data (Figure 1.9). For example, we can use the function of day length, the gradient of the day length, or the day-night ratio $\varrho = \lambda/(24 - \lambda)$.

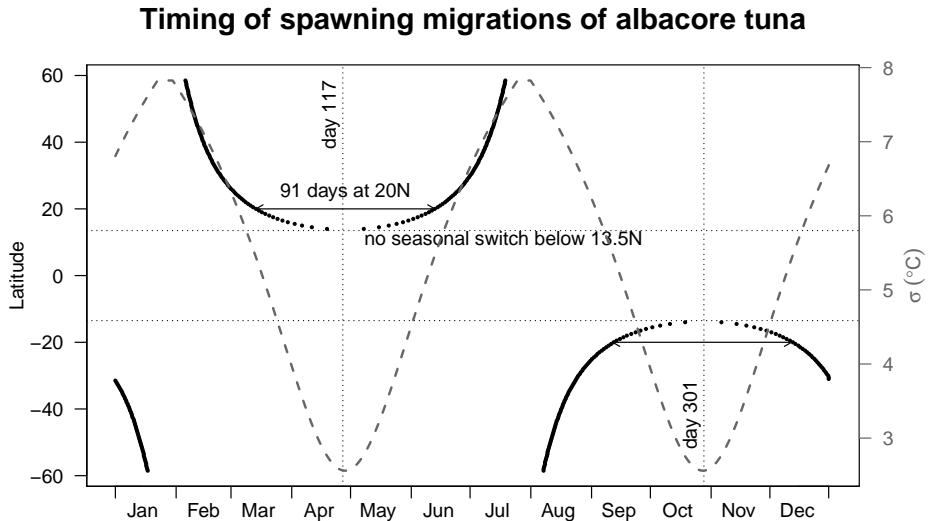


Figure 1.9: Same as in figure 1.8, but including the variations of standard deviation of the Gaussian thermal function (dashed curve, right y-axis), enabling the local cues (expressed through non-zero gradients of seasonal habitat) for mature adult movements. Parameters of seasonal switch estimated based on fisheries data for south Pacific albacore (Senina *et al.*, 2020a).

Local gradients To enable the local cues and hence the non-zero velocities for tunas directed towards the spawning grounds and back to the feeding grounds, the following approach has been implemented. Based on the tuna sensitivity to water temperature and the hypothesis that adult tunas follow warmer temperatures while moving towards maturation

and spawning areas (Le Gall, 1949), we suggest using thermal gradients as the driver of migrations. In SEAPODYM, the temperature is the common environmental variable to which tunas respond within both feeding and spawning habitats. The thermal preferences at all ages $a \in (0, A^+)$ are described by Gaussian-type functions with two parameters, T_a^* and σ_a , being the preferred optimum and the temperature tolerance range respectively (1.17). To link seasonal movements to thermal gradients, we assume that the range of preferred temperatures in the mature tuna's habitat varies with time. It is maximal at the beginning of migrations, that is, at the moment of the switch, and decreases to the range of temperatures characteristic for the spawning at the seasonal peak. One way to describe this variable $\sigma_{a,s}(t)$ during the spawning season is to relate it linearly to the function of day length $\varrho(t, y_n)$ at a given (fixed to the northernmost) latitude y_n , namely to its deviation $\delta\varrho$ from its value at season peak $\varrho(\hat{t}, y_n)$:

$$\sigma_{a,s} = \sigma_0 + (\bar{\sigma}_a - \sigma_0) \frac{\delta\varrho}{\delta\varrho_{max}} \quad (1.35)$$

where σ_0 is the standard deviation providing a range of preferred temperatures in spawning habitat; $\bar{\sigma}_a = T_0^* - T_a^*$ is the difference between optimal spawning temperatures and preferred temperatures of tunas aged a ; and $\delta\varrho_{max}$ is the maximum deviation of the chosen day length functions. Note that one standard deviation $\sigma_{a,s}$ being the difference between H_a and H_s thermal optima provides the highest gradients of the Gaussian thermal function. The seasonality of the spawning migrations and variable thermal range $\sigma_{a,s}$ are illustrated in Figure 1.9. Note that the variability of $\sigma_{a,s}$ is effective only within a spawning season occurring once per year in each hemisphere.

1.2.6 Natural mortality

The natural mortality of fish includes two mechanisms – population decay due to the predation process and population abundance decrease due to senescence of individuals with age, denoted $m_P(a)$ and $m_S(a)$, respectively. They are modelled in SEAPODYM by the following equations:

$$m_P(a) = \bar{m}_P e^{-\beta_P a} \quad (1.36)$$

$$m_S(a) = \bar{m}_S a^{\beta_S} \quad (1.37)$$

where parameter $\bar{m}_P > 0$ is the maximal predation rate, corresponding to age 0 and $\bar{m}_S > 0$ is the senescence mortality rate at $a = 1$, and $\beta_P > 0$ and $\beta_S > 0$ define the rates of mortality change with age; their positiveness corresponds to an assumption that predation mortality decreases with age and senescence increases. So the total mortality rate is simply a sum of two terms, predation and senescence mortality (Figure 1.10):

$$m(a) = m_P(a) + m_S(a) \quad (1.38)$$

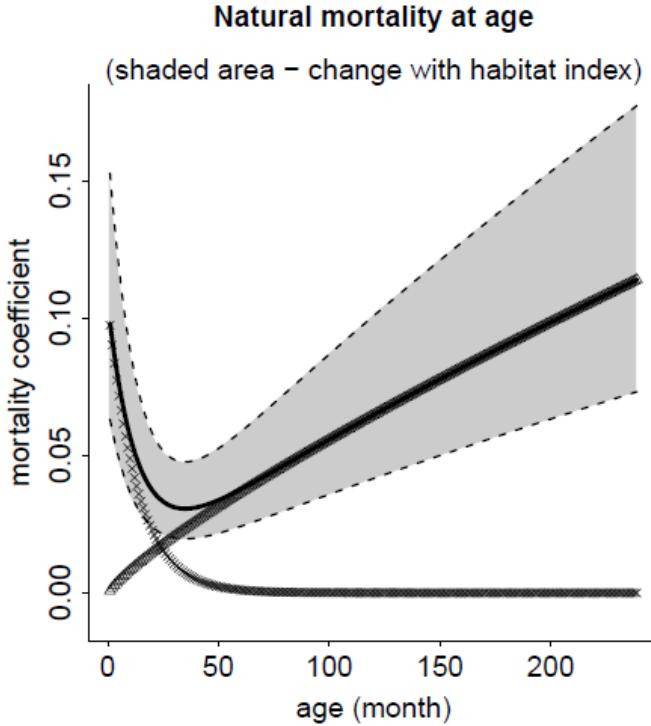


Figure 1.10: Total mortality rates due to senescence, $m_S(a)$ eq. (1.37), due to predation, $m_P(a)$ eq. (1.36), the total mortality rate, $m(a)$ eq. (1.38), as well as the range of variability of mortality rate m_N , eq. (1.40), related to feeding habitat index.

In addition, it is assumed that at any age, the natural mortality can be altered by environmental conditions, such as temperature, availability of food and predators. Hence, we assume that total mortality rate m can vary in space and time depending either on the spawning habitat index H_s (for larval and small juvenile stages) or feeding habitat index H_a for adult life stages. Let us denote I the index of habitat occupied by the life stage:

$$I = \begin{cases} H_s, & \text{if } a = 0 \\ H_j, & \text{if } a \in (0, a_J] \\ H_a, & \text{if } a > a_J \end{cases} \quad (1.39)$$

where the index H_j , used to describe the conditions for survival of juveniles and affecting only the local (in every position \mathbf{x}) mortality rates of juveniles, is computed as follows:

- $H_j = f_1(\Lambda, \alpha) \times f_2(T_1; T_0^*, \sigma_0)$, or
- same as above but including the density-dependent function accounting for cannibalism by adults. However, this is deliberately left out of the scope of this manual due to weak observability of the mechanism from fisheries data.

It is important to note that index H_j does not affect movements of juveniles, hence their distribution can be modified by H_j only in case of strong environmental variability and the variable mortality effect estimated by the MLE approach (see chapter 4).

To account for the effect of environmental conditions on local mortality rates, the mortality at age m in eq. 1.38 is affected by the habitat index variability in time and space. Hence the final formulation of a species' natural mortality $m_N = m_N(a, t, \mathbf{x})$ becomes:

$$m_N = m(a)(1 + \varepsilon)^{1-2I} \quad (1.40)$$

where ε is the parameter that gives the range of variability of mortality rate with habitat index within the range $m_N \in [m(a)/(1+\varepsilon), m(a)(1+\varepsilon)]$ where the minimal value corresponds to the most favourable habitat, $I = 1$, and the maximal to the less favourable habitat, $I = 0$.

1.2.7 Fishing mortality

In SEAPODYM, the mortality due to fishing $m_F = m_F(t, a, \mathbf{x})$ is computed based on two methods: 1) using the Gordon–Schaefer formula and 2) using the catch removal method. The first classical approach relies on the geo-referenced observed fishing effort $E = E(t, \mathbf{x})$ as in the continuous Gordon–Schaefer model, and in the context of age explicit modelling, the mortality rate caused by fishing activity for the part of population at age a is

$$m_F(a) = q \cdot E \cdot s(a), \quad (1.41)$$

where q is the catchability of the fishing gear, and $s(a)$ is its selectivity for the fish at age a . Note, the fish population is usually harvested with different fishing gears, meaning that the model must consider multiple fisheries, so the catchability coefficient, q_f , fishing effort, E_f , as well as the gear selectivity, $s_f(a)$, are all fishery dependent. Then the total fishing mortality is computed as a sum of all mortality rates for fisheries being considered:

$$m_F(a) = \sum_f (q_f \cdot E_f \cdot s_f(a)). \quad (1.42)$$

The selectivity, $s_f(a)$, being a function of age (length), is computed either as a non-linear concave function with a limit one (type I selectivity function), sigmoid function (type II selectivity function) or asymmetric Gaussian (type III):

$$s_f(a) = \begin{cases} \frac{\ell(a)}{\varsigma_f + \ell(a)}, & \text{type I} \\ \left(1 + e^{-\varsigma_f(\ell(a) - \hat{l}_f)}\right)^{-1}, & \text{type II} \\ e^{-\frac{(\ell(a) - \hat{l}_f)^2}{\sigma_{s_f}}}, & \text{if } \ell(a) \leq \hat{l}, \text{ type II,} \\ \mu_f + (1 - \mu_f)e^{-\frac{(\ell(a) - \hat{l}_f)^2}{\sigma_{s_f}}}, & \text{if } \ell(a) > \hat{l}, \text{ type III.} \end{cases} \quad (1.43)$$

where

- ς_f is a half-saturation constant in the selectivity function type I;
- ς_f is a slope coefficient in sigmoid-shape selectivity;
- \hat{l}_f and σ_{s_f} are the parameters of Gaussian function (mean and standard deviation); and
- μ_f is the asymptotic value of selectivity at large fish sizes.

The choice of the functional form of selectivity depends on fishing gear and hence is determined by different factors such as the use of hooks or seine, their depths and mesh size (in case of seine). For example, the sigmoid function is commonly used for long-line fishery, while for purse seine gear, operating at the surface and targeting mostly the younger (smaller) fish, type III is more appropriate (see Figure 1.11).

The second method, called the *catch removal* method, consists of subtraction of total (summarised over all fisheries) observed catch in number of fish of a given age class directly from the predicted fish density. Note that since catch is by definition a discrete variable, it is more convenient to use discrete notations for this variable as well as for all operators including it. Obviously, catch subtraction occurs only in those locations in space where catch is non-zero. Let us denote all variables in a position of a discretised space, that is, in a grid cell with indices (i, j) . Subscripts f, K, p, i, j denote fishery, model integration time step index, age index and grid cell indices respectively (notations of Chapter 2 **Discretisation and numerical solution**).

Let us denote by $N_{k,p,i,j}$ the population density in number of fish within age class p per unit area at the model integration sub-step $k \frac{\Delta T}{n_t}$ ($k = 1, \dots, n_t$, n_t – number of iterations in the inner loop of ADI method). Quantity $C_{k,p,i,j}^{\text{obs}} = \frac{1}{n_t} C_{K,p,i,j}^{\text{obs}}$ is the n th portion of the total observed catch in number of individuals from age class p and time interval K . The fishing mortality is accounted through the subtraction (removal) of the observed catch directly from the population density at the ADI method sub-step. To ensure the positivity of the fish population density, it can only be reduced by the amount that corresponds to the sustained catch, that is smaller than the model biomass in a given position in space. Hence, we have the following:

$$N_{k+1,p,i,j} = N_{k,p,i,j} - \min \left(\frac{C_{k,p,i,j}^{\text{obs}}}{\Delta x_i \Delta y}, N_{k,p,i,j} \right), \quad (1.44)$$

where a differentiable version of the min function is implemented using a rotated hyperbola in order to avoid a problem of the non-differentiability of the minimum of two values.

Note that the age stratification in catch is usually not available in the observational datasets, but the fraction of observed total catch by age $C_{p,i,j}^{\text{obs}}$ can be derived either from

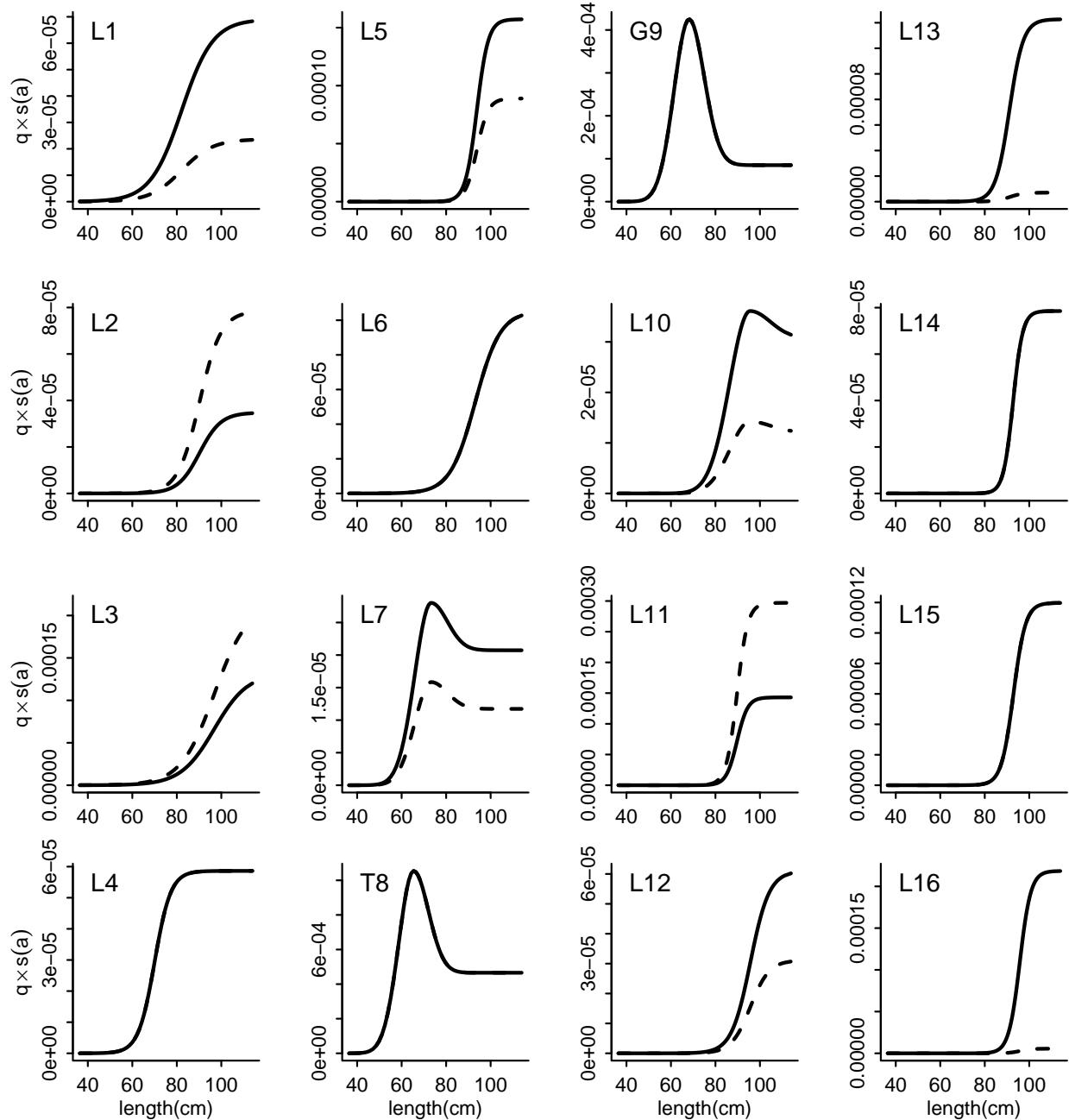


Figure 1.11: Catchability and selectivity by fishery estimated for different optimization experiments for south Pacific albacore (from Senina *et al.*, 2020a). The y-axis shows the product of catchability coefficient (constant in space and time) and selectivity function (varies with size between 0 and 1) so that the plot gives catchability by size. For fisheries for which a dashed line is present, the catchability was allowed to vary linearly in time to account for the change in the gears/strategy efficiency. In that case the dashed lines correspond to the catchability at size at the beginning of the run and the solid lines show the catchability at size at the end of the run.

the observed length frequency distributions, or simply using the model fishery selectivity function. Because of scarcity of fine-resolution length frequency data, currently the second method is implemented. The latter means that in the catch removal method even though we use the observed quantities, the mortality induced by a fishery depends on model parameters. For more details on catch prediction and parameter estimation see chapter 4 **Model parametrisation**.

1.3 Model parameters

We summarise in Table 1.3 all model parameters that control the SEAPODYM dynamic processes described in this chapter. Note that biological parameters that are used to compute species' length-at-age, weight-at-age as well as maturity-at-age (section 1.2.4), are not included in this table as they are not SEAPODYM parameters and usually estimated in external studies. As described in section 1.2, all dynamic processes in the model are linked to the environmental variables. In addition, the spatial dynamics (movement) that greatly influences all demographic processes in heterogeneous environments, is modelled explicitly. Therefore, a complex and highly dimensional dynamics can be governed by a limited number of parameters. They are listed in Table 1.3: reproduction parameters controlling spawning habitat, H_s , and source term, S ; five parameters of natural mortality, M ; 12 parameters for feeding habitat, H_a ; and four parameters of movement rates, among which two control advection rates v_N and two define diffusion rates, D . In addition, there are two to four parameters per fishery, which contribute to model dynamics through fishing pressure, the number of them depending on the use of a mortality equation (1.42 or 1.44) and selectivity function eq. 1.43. However, fisheries parameters also influence the model predictions, such as predicted catch-at-age and length frequency of catch. The methods for catch and length frequency data predictions are fully detailed in Chapter 4.

Table 1.3: Species dynamic parameters with their notations, definitions and references

θ	Eq.	Description	Units
<i>Recruitment</i>			
r	(1.13)	Reproduction rate in Beverton–Holt function	mo^{-1}
b	(1.13)	slope parameter in Beverton–Holt function	$\text{Nb}^{-1}\text{km}^2$
<i>Natural mortality</i>			
\bar{m}_p	(1.36)	Predation mortality rate age 0	mo^{-1}
β_p	(1.36)	Slope coefficient in predation mortality	
\bar{m}_s	(1.37)	Senescence mortality rate at age 0	$\text{mo}^{-1-\beta_s}$
β_s	(1.37)	Slope coefficient in senescence mortality	
ϵ	(1.40)	Variability of mortality rate with habitat index from $\frac{M}{(1+\epsilon)}$ in the worst habitat to $M(1+\epsilon)$ in the best habitat	

Table 1.3 continued

θ	Eq.	Description	Units
<i>Spawning habitat</i>			
σ	(1.7)	Standard deviation in temperature Gaussian function of spawning habitat	°C
T^*	(1.7)	Optimal surface temperature for larvae in spawning habitat definition	°C
α	(1.8)	Prey encounter rate in Holling (type III) function	day ⁻¹
α_F	(1.9)	Log-normal mean parameter in predator-dependent function	g·m ⁻²
β_F	(1.9)	Log-normal shape parameter in predator-dependent function	
<i>Feeding habitat</i>			
T_0	(1.19)	Preferred temperature for age 0	°C
T_m	(1.19)	Preferred temperature for the oldest adult fish	°C
σ_0	(1.20)	Standard deviation in temperature Gaussian function for age 0 in feeding habitat	°C
σ_m	(1.20)	Standard deviation in temperature Gaussian function at age A+	°C
b_T	(1.19)	Allometric power coefficient for thermal preferences at age	
γ	(1.18)	Slope of the oxygen function in layer accessibility	
\hat{O}_2	(1.18)	Critical value of dissolved oxygen in the layer	ml/L
E_{11}	(1.21)	Epipelagic forage scaling factor	m ² ·g ⁻¹
E_{22}	(1.21)	Mesopelagic forage scaling factor	m ² ·g ⁻¹
E_{21}	(1.21)	Migrant mesopelagic forage scaling factor	m ² ·g ⁻¹
E_{33}	(1.21)	Lower mesopelagic forage scaling factor	m ² ·g ⁻¹
E_{32}	(1.21)	Migrant lower mesopelagic forage scaling factor	m ² ·g ⁻¹
E_{31}	(1.21)	Highly migrant lower mesopelagic forage scaling factor	m ² ·g ⁻¹
<i>Adult seasonal migrations</i>			
\hat{t}	(1.33)	Mid-date (day of the year) of seasonal spawning migrations of adults	day
ϱ_s	(1.34)	Critical value of day-night ratio, ϱ , triggering seasonal migrations	
<i>Movement</i>			
c	(1.30)	Coefficient of diffusion variability with habitat index	
σ	(1.30)	Multiplier for the theoretical diffusion rate $\frac{\bar{V}^2 \Delta T}{4}$	
V_m	(1.29)	Velocity at maximal habitat gradient and $A = 1$, BL/s	
A	(1.29)	Slope coefficient in allometric function for tuna velocity	
<i>Fishing mortality and catch prediction</i>			
q_f	(1.42, 4.2)	Constant catchability coefficients for all declared fisheries	
ς_f	(1.43)	Steepness of selectivity function if type I or II, standard deviation if type III	cm

Table 1.3 continued

θ	Eq.	Description	Units
\hat{l}_f	(1.43)	Threshold length in sigmoid function; mean length in asymmetric Gaussian function	cm
μ_f	(1.43)	Lowest selectivity for large fish in case of asymmetric Gaussian selectivity function	

References

- Beverton, R., Holt, S. 1957. On the dynamics of exploited fish populations. Gt. Britain, Fishery Invest., Ser. II, Vol. XI. 533pp.
- Bertalanffy, L. von. 1938. A Quantitative Theory of Organic Growth (Inquiries on Growth Laws. II). *Human Biology*, 10:2, 181-213.
- Brill, R. 1994. A review of temperature and oxygen tolerance studies of tunas pertinent to fisheries oceanography, movement models and stock assessments. *Fisheries Oceanography* 3:3, 204-216.
- Cayre, P. 1991. Behavior of yellowfin tuna (*Thunnus albacares*) and skipjack tuna (*Katsuwonus pelamis*) around fish aggregating devices (FADs) in the Comoros Islands as determined by ultrasonic tagging. *Aquat Living Resour* 4-112.
- Cury P. 1994. Obstinate nature: An ecology of individuals: Thoughts on reproductive behavior and biodiversity. *Canadian Journal of Fisheries and Aquatic Sciences* 51: 1664–1673.
- Delpech, A., Conchon, A., Titaud, O., Lehodey, P. 2020. Influence of oceanic conditions in the energy transfer efficiency estimation of a micronekton model. *Biogeosciences*, 17: 833–850. <https://doi.org/10.5194/bg-17-833-2020>
- Holland, K.N., Brill, R.W., Chang, R.K., Sibert, J.R., Fournier, D.A., 1992. Physiological and behavioral thermoregulation in bigeye tuna (*Thunnus obesus*). *Nature* 358, 110–112.
- Iverson, R. L. 1990. Control of marine fish production. *Limnol. Oceanogr.* 35:1593-1604.
- Le Gall, J. 1949. Résumé des connaissances acquises sur la biologie du germon. *Rev. Trav. Off. Pech. marit..* 15:1-42.
- Lehodey, P., Chai, F., Hampton, J. 2003. Modelling climate-related variability of tuna populations from a coupled ocean biogeochemical-populations dynamics model. *Fish. Oceanogr.* 12: 4(5), 483-494.

- Lehodey P., Senina I. and Murtugudde R. 2008. A spatial ecosystem and populations dynamics model (SEAPODYM) — modeling of tuna and tuna-like populations. *Progress in Oceanography* 78: 304–318. <https://doi.org/10.1016/j.pocean.2008.06.004>
- Lehodey P., Conchon A., Senina I., Domokos R., Calmettes B. et al. 2015. Optimization of a micronekton model with acoustic data. *ICES Journal of Marine Research.* 53, 571-607. *Science* 72(5):1399–1412. <https://doi.org/10.1093/icesjms/fsu233>
- Llopiz, J. K., Richardson, D., Shiroza, A., Smith, S., Cowen, R. 2010. Distinctions in the diets and distributions of larval tunas and the important role of appendicularians. *Limnol. Oceanogr.* 55, 983–996.
- Maury, O., 2005. How to model the size-dependent vertical behavior of bigeye (*Thunnus obesus*) tuna in its environment? Col. Vol. Sci. Pap. ICCAT 57 (2), 115–126.
- Nihira, A. 1996. Studies on the behavioral ecology and physiology of migratory fish schools of skipjack tuna (*Katsuwonus pelamis*) in the oceanic frontal area. *Bull. Tohoku Natl. Fish. Res. Inst.* 58, 137-233.
- Senina, I., Lehodey, P., Calmettes, B., Nicol, S., Caillot, S., Hampton, J., Williams, P. 2016. Predicting skipjack tuna dynamics and effects of climate change using SEAPODYM with fishing and tagging data. Document number WCPFC-SC12-2016/EB WP-01. Western and Central Pacific Fisheries Commission.
- Senina, I., Lehodey, P., Calmettes, B., Dessert, M., Nicol, S., Hampton, J., et al. 2018. Impact of climate change on tropical tuna species and tuna fisheries in Pacific Island waters and high seas areas. Document number WCPFCSC14-2018/EB-WP-01. Western and Central Pacific Fisheries Commission.
- Senina, I., Lehodey, P., Hampton, J. and J. Sibert. 2020a. Quantitative modelling of the spatial dynamics of South Pacific and Atlantic albacore tuna populations. *Deep Sea Res. II* 175, doi.org/10.1016/j.dsr2.2019.104667).
- Senina, I., Lehodey, P., Nicol, S., Scutt Phillips, J., Hampton, J., 2020b. SEAPODYM: revisiting bigeye reference model with conventional tagging data. WCPFC-SC16-2020.
- Senina, I., Briand, G., Lehodey, P., Nicol, S., Hampton, J., Williams, P. 2021. Reference model of bigeye tuna using SEAPODYM with catch, length and conventional tagging data. Information paper, 17th Regular Session of the Scientific Committee of the Western Central Pacific Fisheries Commission, SC17-EB-IP-08. <https://meetings.wcpfc.int/node/12605>
- Turchin, P. 1998. Quantitative analysis of movement. Sinauer, Sunderland, MS.

Chapter 2

Discretisation and numerical solution

The SEAPODYM underlying partial differential equations (PDE) with initial and boundary conditions eqs. (1.1-1.3) are discretised on a regular grid, approximated using the finite-difference method and numerically solved with the alternating-direction implicit (ADI) method (Press et al., 1992). The numerical approximation scheme and the numerical ADI solver, which are used in SEAPODYM, were developed and implemented by Sibert and Fournier (1994) for the general ADR model without an ageing term. It is fully described in Sibert and Fournier (1994) and Sibert et al. (1999). The choice of a classical ADI method despite its known drawbacks, for example, in terms of numerical dispersion, was made for its unconditional stability (Press et al., 1992), which ensures convergence to a solution for all step sizes and dynamic rates. This method's property enables integration of the numerical model within the optimization method, which implies variations of model parameters.

The model equations are discretised on a regular Arakawa-A grid. The use of the Arakawa-A grid for the spatial dimensions implies that all quantities are evaluated at the same position, here in the centre of the grid cells. The spatial domain has irregular boundaries. Several boundary conditions are implemented: zero-flux (Neumann) boundary conditions, absorbing (Dirichlet) boundary conditions and conditions for connected east–west borders in the case of a global domain. This chapter describes in detail the discretisation of the model dimensions, the finite-difference approximation of partial derivatives and the numerical solver.

2.1 Discretisation of model dimensions

2.1.1 Space discretisation

The spatial domain $\Omega \in [X_0, X_m] \times [Y_0, Y_m]$ is uniformly discretised in $n_x \times n_y$ cells, with a constant step Δx (resp. Δy) along the x (resp. y) direction, with the origin $(0, 0)$ being the

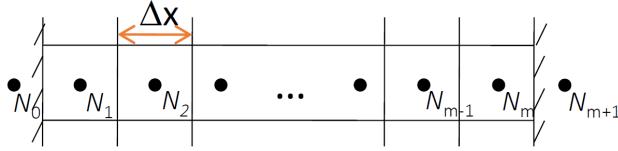


Figure 2.1: Regular Arakawa-A grid used in numerical approximation of SEAPODYM equations. The quantity $N_i = N(x_i, \cdot)$ is the population density described by model eq. (1.1).

north-west corner of the model domain:

$$\Delta x = (X_m - X_0)/n_x, \quad \Delta y = (Y_m - Y_0)/n_y. \quad (2.1)$$

We denote by (i, j) , $i = 1, \dots, n_x$, $j = 1, \dots, n_y$ the cell that corresponds to the subdomain $[X_0 + (i - 1)\Delta x, X_0 + i\Delta x] \times [Y_0 + (j - 1)\Delta y, Y_0 + j\Delta y]$. All quantities are evaluated at (x_i, y_j) , the centre of the cell (i, j) of the Arakawa-A grid (Figure 2.1), that is

$$x_i = X_0 + \left(i - \frac{1}{2}\right)\Delta x, \quad i = 1, \dots, n_x, \quad (2.2)$$

$$y_j = Y_0 + \left(j - \frac{1}{2}\right)\Delta y, \quad j = 1, \dots, n_y. \quad (2.3)$$

2.1.2 Time discretisation

The ADR equation is numerically solved using a multi-incremental algorithm. The time period $[T_0, T_m]$ is first split into n_K steps or **outer loops**:

$$\Delta T = (T_m - T_0)/n_T, \quad T_K = T_0 + K\Delta T, \quad K = 0, \dots, n_T. \quad (2.4)$$

where the coefficients of the ADR equation are supposed to be constant. Each outer loop is in turn discretised into n_t uniform substeps or **inner-loops**: for a given $K \in \{0, n_T - 1\}$ we define

$$\Delta t = (T_{K+1} - T_K)/n_t, \quad t_k = T_K + k\Delta t, \quad k = 0, \dots, n_t. \quad (2.5)$$

This multi-incremental algorithm is applied with an implicit Euler integration scheme (see below) that ensures both stability and good approximation of the solution.

2.1.3 Age discretisation

Population age dimension is discretised into age classes: from the first age class including age $a = 0$ to the last age class, including the maximal age \bar{m} of the individuals in the population. In addition, the last age class, also called the $A+$ class, is larger in size than

the other age classes. Thus, we define n_a age classes characterized by their age intervals $[a_p, a_{p+1}]$, $p = 0, \dots, n_a$ and the age steps (or age class life period) as follows:

$$\Delta a_p = a_{p+1} - a_p, \quad p = 0, \dots, n_a - 1, \quad (2.6)$$

where the age steps $\Delta a_p = \text{const}$ and $\Delta a_{n_a-1} > \Delta a_p$ for $p = 0, \dots, n_a - 2$ specify uniformity of all age classes but the last one. Mean age of the age class (p) is then given as

$$\tilde{a}_p = \frac{a_{p+1} + a_p}{2}, \quad p = 0, \dots, n_a. \quad (2.7)$$

Dimension	Variable	Range	Index	Range	Step
x-coordinate	x_i	$[X_0, X_m]$	i	$1, \dots, n_x$	Δx
y-coordinate	y_j	$[Y_0, Y_m]$	j	$1, \dots, n_y$	Δy
time (outer loop)	T_K	$[T_0, T_m]$	K	$0, \dots, n_T$	ΔT
time (inner loop)	t_k	$[T_K, T_{K+1}]$	k	$0, \dots, n_t$	Δt
age	a_p	$[0, \bar{a}]$	p	$0, \dots, n_a$	Δa_p

Table 2.1: Dimension discretisation indices

Notation convention:

- Index names and their meaning are shown in Table 2.1.
- For a given quantity Q we define

$$Q_{k,p,i,j} = Q(t_k, \tilde{a}_p, x_i, y_j), \quad (2.8)$$

and

$$Q_{K,p,i,j} = Q(T_K, \tilde{a}_p, x_i, y_j). \quad (2.9)$$

- When there is no loss of clarity we may omit dependence of some variables (or indices).

2.2 Approximation of equations and numerical scheme

The partial derivatives of system (1.1-1.3) are approximated using centred finite differences. Time integration is performed using a two-step splitting method: advection-diffusion and mortality processes are solved using ADI method along spatial directions: $k \rightarrow k + \frac{1}{2}$ for the x -direction, $k + \frac{1}{2} \rightarrow k + 1$ for the y -direction. The ageing term is integrated outside of the time integration loop.

During an outer loop period $[T_{K-1}, T_K]$, we consider that the forcings terms (parameters of advection-diffusion and mortality processes) are constant and equal to their value at T_K in order to be consistent with the implicit scheme.

2.2.1 Age integration

The ageing process is taken into account in between the time integration loops once the solution of the ADR equation is computed over the full time step, that is, between the time steps $[T_{K-1}, T_K]$ and $[T_K, T_{K+1}]$ (see also section 2.4). The integration of the ageing term is then done by updating the cohort densities as follows:

$$\begin{cases} N_{K,n_a} &= N_{K,n_a} + N_{K,n_a-1} \\ N_{K,p} &= N_{K,p-1} \end{cases} \quad \text{for } p = (n_a - 1) \rightarrow 1 \quad (2.10)$$

2.2.2 Time integration

Time integration from time T_{K-1} to time T_K in the ADI method is done using an iterative **two-step splitting method along spatial directions**. During these two steps, the ageing process $\partial_a N$ is not taken into account, that is, age class index p is kept constant during time integration.

First half step

The first implicit half inner-loop time step $k \rightarrow k + \frac{1}{2}$ along direction x solves for each inner-loop index $k = 0 \dots n_t$:

$$\frac{N_{k+\frac{1}{2}} - N_k}{\Delta t/2} - \frac{\partial}{\partial x} \left(D_K \frac{\partial N_{k+\frac{1}{2}}}{\partial x} \right) + \frac{\partial}{\partial x} (u_K N_{k+\frac{1}{2}}) + M_K N_{k+\frac{1}{2}} = \frac{\partial}{\partial y} \left(D_K \frac{\partial N_k}{\partial y} \right) - \frac{\partial}{\partial y} (v_K N_k). \quad (2.11)$$

Second half step

The second implicit half inner-loop time step $k + \frac{1}{2} \rightarrow k + 1$ along direction y solves for each inner-loop index $k = 0 \dots n_t$:

$$\frac{N_{k+1} - N_{k+\frac{1}{2}}}{\Delta t/2} - \frac{\partial}{\partial y} \left(D_K \frac{\partial N_{k+1}}{\partial y} \right) + \frac{\partial}{\partial y} (v_K N_{k+1}) = \frac{\partial}{\partial x} \left(D_K \frac{\partial N_{k+\frac{1}{2}}}{\partial x} \right) - \frac{\partial}{\partial x} (u_K N_{k+\frac{1}{2}}) - M_K N_{k+\frac{1}{2}}. \quad (2.12)$$

We then apply discretisation operators (2.13), (2.14), (2.17) and (2.18) depending on the considered cell. For clarity we omit the age class index p in the following paragraphs.

2.2.3 Partial derivatives approximation

Hereafter we denote by u and v the components of the total velocity $\mathbf{v} = \mathbf{v}_c + \mathbf{v}_N$ in eq. (1.1).

Advection terms

Derivatives $\frac{\partial}{\partial x}(uN)$ and $\frac{\partial}{\partial x}(vN)$ are approximated by centred upwind differencing:

$$\left. \frac{\partial(uN)}{\partial x} \right|_{i,j} \approx \begin{cases} \frac{1}{\Delta x}(u_{i,j}N_{i,j} - u_{i-1,j}N_{i-1,j}) & \text{if } u_{i,j} > 0, \\ \frac{1}{\Delta x}(u_{i+1,j}N_{i+1,j} - u_{i,j}N_{i,j}) & \text{if } u_{i,j} < 0, \end{cases} \quad (2.13)$$

$$\left. \frac{\partial(vN)}{\partial y} \right|_{i,j} \approx \begin{cases} \frac{1}{\Delta y}(v_{i,j}N_{i,j} - v_{i,j-1}N_{i,j-1}) & \text{if } v_{i,j} > 0, \\ \frac{1}{\Delta y}(v_{j+1,j}N_{i,j+1} - v_{i,j}N_{i,j}) & \text{if } v_{i,j} < 0. \end{cases} \quad (2.14)$$

Diffusion terms

Derivatives $\frac{\partial}{\partial x}\left(D\frac{\partial N}{\partial x}\right)$ and $\frac{\partial}{\partial y}\left(D\frac{\partial N}{\partial y}\right)$ are approximated using centred differences evaluated as the mean between the forward finite difference and the backward finite difference of the corresponding terms. We get:

$$\left. \frac{\partial}{\partial x} \left(D \frac{\partial N}{\partial x} \right) \right|_{i,j} = \left. \frac{\partial D}{\partial x} \right|_{i,j} \times \left. \frac{\partial N}{\partial x} \right|_{i,j} + D_{i,j} \frac{\partial^2 N}{\partial x^2} \quad (2.15)$$

$$\begin{aligned} &\approx \frac{1}{2} \left(\frac{D_{i+1,j} - D_{i,j}}{\Delta x} \times \frac{N_{i+1,j} - N_{i,j}}{\Delta x} \right. \\ &\quad \left. + \frac{D_{i,j} - D_{i-1,j}}{\Delta x} \times \frac{N_{i,j} - N_{i-1,j}}{\Delta x} \right) \\ &\quad + D_{i,j} \times \frac{N_{i-1,j} - 2N_{i,j} + N_{i+1,j}}{\Delta x^2} \end{aligned} \quad (2.16)$$

$$\begin{aligned} &\approx N_{i-1,j} \frac{D_{i,j} + D_{i-1,j}}{2\Delta x^2} - N_{i,j} \frac{D_{i+1,j} + 2D_{i,j} + D_{i-1,j}}{2\Delta x^2} \\ &\quad + N_{i+1,j} \frac{D_{i,j} + D_{i+1,j}}{2\Delta x^2} \end{aligned} \quad (2.17)$$

$$\begin{aligned} \left. \frac{\partial}{\partial y} \left(D \frac{\partial N}{\partial y} \right) \right|_{i,j} &\approx N_{i,j-1} \frac{D_{i,j} + D_{i,j-1}}{2\Delta y^2} - N_{i,j} \frac{D_{i,j+1} + 2D_{i,j} + D_{i,j-1}}{2\Delta y^2} \\ &\quad + N_{i,j+1} \frac{D_{i,j} + D_{i,j+1}}{2\Delta y^2}. \end{aligned} \quad (2.18)$$

2.2.4 Numerical scheme

For a given grid cell inside the model domain (interior cell), we consider four types of neighbours:

- ocean cell inside the computation domain,
- ocean cell outside the computation domain,
- ocean cell within the East–West overlap zone (in the global domain set-up only),
- land cell.

Hence, the first three types of interior grid cells are open cells and the last one is the closed boundary cell. Hereafter we provide the model numerical scheme in the interior grid cells; depending whether they are open or closed cells.

Discretised equations in open grid cells

In the interior cells surrounded by ocean cells the implicit scheme of eq. (2.11) for the first half time step $k \rightarrow k + \frac{1}{2}$ uses spatial terms discretised in the x direction. According to upwind differencing of advection terms, these discrete equations account for the positiveness of $u_{K,i,j}$ and $v_{K,i,j}$:

- Case $u_{K,i,j} > 0$ and $v_{K,i,j} > 0$:

$$\begin{aligned}
& N_{k+\frac{1}{2},i-1,j} \left(-\frac{u_{K,i-1,j}}{\Delta x} - \frac{D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} \right) \\
& + N_{k+\frac{1}{2},i,j} \left(\frac{2}{\Delta t} + \frac{u_{K,i,j}}{\Delta x} + \frac{D_{K,i+1,j} + 2D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} + M_{K,i,j} \right) \\
& + N_{k+\frac{1}{2},i+1,j} \left(-\frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta x^2} \right) = \\
& N_{k,i,j-1} \left(\frac{D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} + \frac{v_{K,i,j-1}}{\Delta y} \right) \\
& + N_{k,i,j} \left(\frac{2}{\Delta t} - \frac{v_{K,i,j}}{\Delta y} - \frac{D_{K,i,j+1} + 2D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) + N_{k,i,j+1} \left(\frac{D_{K,i,j} + D_{K,i,j+1}}{2\Delta y^2} \right)
\end{aligned} \tag{2.19}$$

- Case $u_{K,i,j} > 0$ and $v_{K,i,j} < 0$:

$$\begin{aligned}
& N_{k+\frac{1}{2},i-1,j} \left(-\frac{u_{K,i-1,j}}{\Delta x} - \frac{D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} \right) \\
& + N_{k+\frac{1}{2},i,j} \left(\frac{2}{\Delta t} + \frac{u_{K,i,j}}{\Delta x} + \frac{D_{K,i+1,j} + 2D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} + M_{K,i,j} \right) \\
& + N_{k+\frac{1}{2},i+1,j} \left(-\frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta x^2} \right) = \\
& N_{k,i,j-1} \left(\frac{D_{K,i,j} + D_{K,i,j+1}}{2\Delta y^2} \right) + N_{k,i,j} \left(\frac{2}{\Delta t} + \frac{v_{K,i,j}}{\Delta y} - \frac{D_{K,i,j+1} + 2D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\
& + N_{k,i,j+1} \left(\frac{D_{K,i,j} + D_{K,i,j+1}}{2\Delta y^2} - \frac{v_{K,i,j+1}}{\Delta y} \right) \tag{2.20}
\end{aligned}$$

- Case $u_{K,i,j} < 0$ and $v_{K,i,j} > 0$:

$$\begin{aligned}
& N_{k+\frac{1}{2},i-1,j} \left(-\frac{D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} \right) \\
& + N_{k+\frac{1}{2},i,j} \left(\frac{2}{\Delta t} - \frac{u_{K,i,j}}{\Delta x} + \frac{D_{K,i+1,j} + 2D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} + M_{K,i,j} \right) \\
& + N_{k+\frac{1}{2},i+1,j} \left(\frac{u_{K,i+1,j}}{\Delta x} - \frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta x^2} \right) = \\
& N_{k,i,j-1} \left(\frac{D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} + \frac{v_{K,i,j-1}}{\Delta y} \right) \\
& + N_{k,i,j} \left(\frac{2}{\Delta t} - \frac{v_{K,i,j}}{\Delta y} - \frac{D_{K,i,j+1} + 2D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) + N_{k,i,j+1} \left(\frac{D_{K,i,j} + D_{K,i,j+1}}{2\Delta y^2} \right) \tag{2.21}
\end{aligned}$$

- Case $u_{K,i,j} < 0$ and $v_{K,i,j} < 0$:

$$\begin{aligned}
& N_{k+\frac{1}{2},i-1,j} \left(-\frac{D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} \right) \\
& + N_{k+\frac{1}{2},i,j} \left(\frac{2}{\Delta t} - \frac{u_{K,i,j}}{\Delta x} + \frac{D_{K,i+1,j} + 2D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} + M_{K,i,j} \right) \\
& + N_{k+\frac{1}{2},i+1,j} \left(+\frac{u_{K,i+1,j}}{\Delta x} - \frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta x^2} \right) = \\
& N_{k,i,j-1} \left(\frac{D_{K,i,j} + D_{K,i,j+1}}{2\Delta y^2} \right) + N_{k,i,j} \left(\frac{2}{\Delta t} + \frac{v_{K,i,j}}{\Delta y} - \frac{D_{K,i,j+1} + 2D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\
& + N_{k,i,j+1} \left(\frac{D_{K,i,j} + D_{K,i,j+1}}{2\Delta y^2} - \frac{v_{K,i,j+1}}{\Delta y} \right). \tag{2.22}
\end{aligned}$$

For the second half time step $k + \frac{1}{2} \rightarrow k$, the implicit scheme of eq. (2.12) uses spatial derivatives approximated along the y direction and accounts for the positiveness of $u_{K,i,j}$ and $v_{K,i,j}$

- Case $v_{K,i,j} > 0$ and $u_{K,i,j} > 0$:

$$\begin{aligned}
& N_{k+1,i,j-1} \left(-\frac{v_{K,i,j-1}}{\Delta y} - \frac{D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\
& + N_{k+1,i,j} \left(\frac{2}{\Delta t} + \frac{v_{K,i,j}}{\Delta y} + \frac{D_{K,i,j+1} + 2D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\
& + N_{k+1,i,j+1} \left(-\frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta y^2} \right) = \\
& N_{k+\frac{1}{2},i-1,j} \left(\frac{D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} + \frac{u_{K,i-1,j}}{\Delta x} \right) \\
& + N_{k+\frac{1}{2},i,j} \left(\frac{2}{\Delta t} - \frac{u_{K,i,j}}{\Delta x} - \frac{D_{K,i+1,j} + 2D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} - M_{K,i,j} \right) \\
& + N_{k+\frac{1}{2},i+1,j} \left(\frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta x^2} \right)
\end{aligned} \tag{2.23}$$

- Case $v_{K,i,j} > 0$ and $u_{K,i,j} < 0$:

$$\begin{aligned}
& N_{k+1,i,j-1} \left(-\frac{v_{K,i,j-1}}{\Delta y} - \frac{D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\
& + N_{k+1,i,j} \left(\frac{2}{\Delta t} + \frac{v_{K,i,j}}{\Delta y} + \frac{D_{K,i,j+1} + 2D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\
& + N_{k+1,i,j+1} \left(-\frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta y^2} \right) = \\
& N_{k+\frac{1}{2},i-1,j} \left(\frac{D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} \right) \\
& + N_{k+\frac{1}{2},i,j} \left(\frac{2}{\Delta t} + \frac{u_{K,i,j}}{\Delta x} - \frac{D_{K,i+1,j} + 2D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} - M_{K,i,j} \right) \\
& + N_{k+\frac{1}{2},i+1,j} \left(\frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta x^2} - \frac{u_{K,i+1,j}}{\Delta x} \right)
\end{aligned} \tag{2.24}$$

- Case $v_{K,i,j} < 0$ and $u_{K,i,j} > 0$:

$$\begin{aligned}
& N_{k+1,i,j-1} \left(-\frac{D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\
& + N_{k+1,i,j} \left(\frac{2}{\Delta t} - \frac{v_{K,i,j}}{\Delta y} + \frac{D_{K,i,j+1} + 2D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\
& + N_{k+1,i,j+1} \left(-\frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta y^2} + \frac{v_{K,i,j+1}}{\Delta y} \right) = \\
& N_{k+\frac{1}{2},i-1,j} \left(\frac{D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} + \frac{u_{K,i-1,j}}{\Delta x} \right) \\
& + N_{k+\frac{1}{2},i,j} \left(\frac{2}{\Delta t} - \frac{u_{K,i,j}}{\Delta x} - \frac{D_{K,i+1,j} + 2D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} - M_{K,i,j} \right) \\
& + N_{k+\frac{1}{2},i+1,j} \left(\frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta x^2} \right)
\end{aligned} \tag{2.25}$$

- Case $v_{K,i,j} < 0$ and $u_{K,i,j} < 0$:

$$\begin{aligned}
& N_{k+1,i,j-1} \left(-\frac{D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\
& + N_{k+1,i,j} \left(\frac{2}{\Delta t} - \frac{v_{K,i,j}}{\Delta y} + \frac{D_{K,i,j+1} + 2D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\
& + N_{k+1,i,j+1} \left(-\frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta y^2} + \frac{v_{K,i,j+1}}{\Delta y} \right) = \\
& N_{k+\frac{1}{2},i-1,j} \left(\frac{D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} \right) \\
& + N_{k+\frac{1}{2},i,j} \left(\frac{2}{\Delta t} + \frac{u_{K,i,j}}{\Delta x} - \frac{D_{K,i+1,j} + 2D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} - M_{K,i,j} \right) \\
& + N_{k+\frac{1}{2},i+1,j} \left(\frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta x^2} - \frac{u_{K,i+1,j}}{\Delta x} \right)
\end{aligned} \tag{2.26}$$

Discretised equations in closed grid cells

We now write down the model numerical scheme in closed grid cells, that is, neighbouring with either a land or ocean grid cell assigned as a boundary cell considering three types of boundary conditions: 1) Neumann boundary conditions, 2) Dirichlet boundary conditions, and 3) a special case of Dirichlet boundary conditions in the case of the global domain.

1) Neumann boundary conditions

Neumann boundary conditions specify impermeability of domain boundaries. These conditions are applied in the case of closed cells, that is, neighbouring with either land cells or ocean cells, for which impenetrability can be assumed due to natural causes.

Let's use the case of a *left-closed boundary in the x-direction and open boundaries in the y-direction* in eq. (2.19): we have $N_{k+\frac{1}{2},i-1,j} = N_{k+\frac{1}{2},i,j}$, $D_{K,i-1,j} = 0$ and $u_{K,i-1,j} = 0$. Then if $u_{K,i,j} > 0$ and $v_{K,i,j} > 0$ we get

$$\begin{aligned} N_{k+\frac{1}{2},i,j} \left(\frac{2}{\Delta t} + \frac{u_{K,i,j}}{\Delta x} + \frac{D_{K,i+1,j} + D_{K,i,j}}{2\Delta x^2} + M_{K,i,j} \right) + N_{k+\frac{1}{2},i+1,j} \left(-\frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta x^2} \right) = \\ N_{k,i,j-1} \left(\frac{D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} + \frac{v_{K,i,j-1}}{\Delta y} \right) + N_{k,i,j} \left(\frac{2}{\Delta t} - \frac{v_{K,i,j}}{\Delta y} - \frac{D_{K,i,j+1} + 2D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\ + N_{k,i,j+1} \left(\frac{D_{K,i,j} + D_{K,i,j+1}}{2\Delta y^2} \right) \end{aligned} \quad (2.27)$$

Also if $u_{K,i,j} < 0$ and $v_{K,i,j} > 0$ we get

$$\begin{aligned} N_{k+\frac{1}{2},i,j} \left(\frac{2}{\Delta t} + \frac{D_{K,i+1,j} + D_{K,i,j}}{2\Delta x^2} + M_{K,i,j} \right) + N_{k+\frac{1}{2},i+1,j} \left(\frac{u_{K,i+1,j}}{\Delta x} - \frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta x^2} \right) = \\ N_{k,i,j-1} \left(\frac{D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} + \frac{v_{K,i,j-1}}{\Delta y} \right) + N_{k,i,j} \left(\frac{2}{\Delta t} - \frac{v_{K,i,j}}{\Delta y} - \frac{D_{K,i,j+1} + 2D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\ + N_{k,i,j+1} \left(\frac{D_{K,i,j} + D_{K,i,j+1}}{2\Delta y^2} \right). \end{aligned} \quad (2.28)$$

2) Dirichlet boundary conditions

Dirichlet conditions specify the value that a model variable should take at the boundary. This value can be zero (absorbing boundary) to imply the loss of quantity, or non-zero to account for the incoming quantity from outside the domain. Dirichlet boundary conditions are applied in the case of a regional domain.

Let us take the case of a *left-open-to-global boundary in the x-direction and open boundaries in the y-direction* in eq. (2.19): we have $N_{k+\frac{1}{2},i-1,j} = 0$ and the corresponding contribution is part of the second member of the equation. For example, if $u_{K,i,j} > 0$ and $v_{K,i,j} > 0$ we have

$$\begin{aligned} N_{k+\frac{1}{2},i,j} \left(\frac{2}{\Delta t} + \frac{u_{K,i,j}}{\Delta x} + \frac{D_{K,i+1,j} + 2D_{K,i,j} + D_{K,i-1,j}}{2\Delta x^2} + M_{K,i,j} \right) \\ + N_{k+\frac{1}{2},i+1,j} \left(-\frac{D_{K,i,j} + D_{K,i+1,j}}{2\Delta x^2} \right) = \\ N_{k,i,j-1} \left(\frac{D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} + \frac{v_{K,i,j-1}}{\Delta y} \right) + N_{k,i,j} \left(\frac{2}{\Delta t} - \frac{v_{K,i,j}}{\Delta y} - \frac{D_{K,i,j+1} + 2D_{K,i,j} + D_{K,i,j-1}}{2\Delta y^2} \right) \\ + N_{k,i,j+1} \left(\frac{D_{K,i,j} + D_{K,i,j+1}}{2\Delta y^2} \right). \end{aligned} \quad (2.29)$$

3) Global domain

In case of a global domain, namely, the domain covering 360° in a longitudinal direction, we need to take into account fluxes that come through the western (eastern) boundary from the east (west) of the domain. A special case of *cyclic east–west boundary conditions* is implemented in the SEAPODYM numerical scheme. To make sure that the quantity flows through the open east–west boundaries in the context of the ADI method, we extend the computational domain on both sides along the x (longitudinal) direction to create the buffer zone. The eastern (western) buffer zone is then filled with values from the west (east) of the domain. The size of the buffer zone is defined as large enough to cover the maximal distance of the mass movement during one outer time step $[T_{K-1}, T_K]$. The discretised equation is then solved on this extended domain, while at the end of the resolution, we only keep the solution over the non-extended part of the domain.

2.3 Tridiagonal matrix system and its resolution

The above numerical scheme eqs. (2.19–2.29) can be rewritten in matrix form with the matrices on the left-hand side being tridiagonal. For clarity, we use all dimensional notations here, even though the age index p remains constant within the resolution of linear system. Thus, to get the numerical solution of model (1.1) at a given time step, we solve iteratively (inner loop) two systems of linear equations, and each iteration implies resolution of $n_a \times n_y$ and $n_a \times n_x$ systems of linear equations in a two-step time splitting ADI method.

Thus, at the first half-step we solve $n_a \times n_y$ linear systems of size $n_x \times n_x$ along the x -direction:

$$\mathbf{A}_{K,p,j} \cdot \mathbf{N}_{k+\frac{1}{2},p,j} = \mathbf{g}_{k,p,j} \quad (2.30)$$

and then at the second half-step we solve $n_a \times n_x$ linear systems of size $n_y \times n_y$ along the y -direction:

$$\mathbf{B}_{K,p,i} \cdot \mathbf{N}_{k+1,p,i} = \mathbf{h}_{k+\frac{1}{2},p,i}. \quad (2.31)$$

Vectors $\mathbf{N}_{k+\frac{1}{2},p,j} = (N_{k+\frac{1}{2},p,1,j}, \dots, N_{k+\frac{1}{2},p,n_x,j})^T$ and $\mathbf{N}_{k+1,p,i} = (N_{k+1,p,i,1}, \dots, N_{k+1,p,i,n_y})^T$ consist of unknown values of population density along the x and y dimensions respectively. These values are found by solving these linear systems with the help of the forward–backward Gauss elimination (LU decomposition) method.

The right-hand-side vectors $\mathbf{g}_{k,p,j} = (g_{k,p,1,j}, \dots, g_{k,p,n_x,j})^T$ in system (2.30), and $\mathbf{h}_{k+\frac{1}{2},p,i} = (h_{k+\frac{1}{2},p,i,1}, \dots, h_{k+\frac{1}{2},p,i,n_y})^T$ in system (2.31) are constructed either from previous step solution, at $K - 1$ (for $k = 0$), or from an intermediate solution of the Gauss method, at all

$k = 1, \dots, n_t$, obtained after each iteration of inner loop of the ADI method. Thus, the elements of vector $\mathbf{g}_{k,p,j}$ are computed for given $a = 1, \dots, n_a$ and $j = 1, \dots, n_y$ from the coefficients of the matrix \mathbf{B} and density at time k as follows:

$$g_{k,p,i,j} = -d_{K,p,i,j} N_{k,p,i,j-1} + \left(\frac{4}{\Delta t} - e_{K,p,i,j} \right) N_{k,p,i,j} - f_{k,p,i,j} N_{k,p,i,j+1}. \quad (2.32)$$

The elements of vector $\mathbf{h}_{k+\frac{1}{2},p,i}$ are computed for given $p = 0, \dots, n_a$ and $i = 1, \dots, n_x$ from the coefficients of the matrix \mathbf{A} and density at time $k + \frac{1}{2}$ as follows:

$$h_{k+\frac{1}{2},p,i,j} = -a_{K,p,i,j} N_{k+\frac{1}{2},p,i-1,j} + \left(\frac{4}{\Delta t} - b_{K,p,i,j} \right) N_{k+\frac{1}{2},p,i,j} - c_{K,p,i,j} N_{k+\frac{1}{2},p,i+1,j}. \quad (2.33)$$

Matrix \mathbf{A} of the linear system (2.30) is defined for all $p = 1, \dots, n_a$ and for all $j = 1, \dots, n_y$ as follows (notations of Sibert and Fournier, 1994):

$$\mathbf{A}_{K,p,j} = \begin{pmatrix} b_{K,p,1,j} & c_{K,p,1,j} & 0 & \dots & 0 & 0 \\ a_{K,p,2,j} & b_{K,p,2,j} & c_{K,p,2,j} & 0 & \dots & 0 \\ 0 & a_{K,p,3,j} & b_{K,p,3,j} & c_{K,p,3,j} & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & a_{K,p,n_x-1,j} & b_{K,p,n_x-1,j} & c_{K,p,n_x-1,j} \\ 0 & \dots & \dots & 0 & a_{K,p,n_x,j} & b_{K,p,n_x,j} \end{pmatrix}. \quad (2.34)$$

Matrix \mathbf{B} of the linear system (2.31) is defined for all $p = 1, \dots, n_a$, and for all $i = 1, \dots, n_x$ as follows:

$$\mathbf{B}_{K,p,i} = \begin{pmatrix} e_{K,p,i,1} & f_{K,p,i,1} & 0 & \dots & 0 & 0 \\ d_{K,p,i,2} & e_{K,p,i,2} & f_{K,p,i,2} & 0 & \dots & 0 \\ 0 & d_{K,p,i,3} & e_{K,p,i,3} & f_{K,p,i,3} & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & d_{K,p,i,n_y-1} & e_{K,p,i,n_y-1} & f_{K,p,i,n_y} \\ 0 & \dots & \dots & 0 & d_{K,p,i,n_y} & e_{K,p,i,n_y} \end{pmatrix}. \quad (2.35)$$

Below we provide the diagonal coefficients of matrices \mathbf{A} and \mathbf{B} given the type of the boundary conditions.

2.3.1 Diagonal coefficients for Neumann boundary conditions

For a given $p = 1, \dots, n_a$ and $j = 1, \dots, n_y$, the elements of matrix $\mathbf{A}_{K,p,j}$ are¹

$$a_{K,p,i,j} = \begin{cases} -\frac{u_{K,p,i-1,j}}{\Delta x} - \frac{D_{K,p,i,j} + D_{K,p,i-1,j}}{2\Delta x^2} & \text{open right closed } u_{K,p,i,j} > 0 \\ -\frac{D_{K,p,i,j} + D_{K,p,i-1,j}}{2\Delta x^2} & \text{open right closed } u_{K,p,i,j} < 0 \\ 0 & \text{left closed} \end{cases} \quad (2.36)$$

$$b_{K,p,i,j} = \begin{cases} \frac{2}{\Delta t} + \frac{u_{K,p,i,j}}{\Delta x} + \frac{D_{K,p,i+1,j} + 2D_{K,p,i,j} + D_{K,p,i-1,j}}{2\Delta x^2} \\ + M_{K,p,i,j} & \text{open } u_{K,p,i,j} > 0 \\ \frac{2}{\Delta t} - \frac{u_{K,p,i,j}}{\Delta x} + \frac{D_{K,p,i+1,j} + 2D_{K,p,i,j} + D_{K,p,i-1,j}}{2\Delta x^2} \\ + M_{K,p,i,j} & \text{open } u_{K,p,i,j} < 0 \\ \frac{2}{\Delta t} + \frac{D_{K,p,i,j} + D_{K,p,i-1,j}}{2\Delta x^2} + M_{K,p,i,j} & \text{right closed } u_{K,p,i,j} > 0 \\ \frac{2}{\Delta t} - \frac{u_{K,p,i,j}}{\Delta x} + \frac{D_{K,p,i,j} + D_{K,p,i-1,j}}{2\Delta x^2} + M_{K,p,i,j} & \text{right closed } u_{K,p,i,j} < 0 \\ \frac{2}{\Delta t} + \frac{D_{K,p,i+1,j} + D_{K,p,i,j}}{2\Delta x^2} + M_{K,p,i,j} & \text{left closed } u_{K,p,i,j} < 0 \\ \frac{2}{\Delta t} + \frac{u_{K,p,i,j}}{\Delta x} + \frac{D_{K,p,i+1,j} + D_{K,p,i,j}}{2\Delta x^2} + M_{K,p,i,j} & \text{left closed } u_{K,p,i,j} > 0 \\ \frac{2}{\Delta t} + M_{K,p,i,j} & \text{closed} \end{cases} \quad (2.37)$$

$$c_{K,p,i,j} = \begin{cases} \left(-\frac{D_{K,p,i,j} + D_{K,p,i+1,j}}{2\Delta x^2} \right) & \text{open left closed } u_{K,p,i,j} > 0 \\ \left(\frac{u_{K,p,i+1,j}}{\Delta x} - \frac{D_{K,p,i,j} + D_{K,p,i+1,j}}{2\Delta x^2} \right) & \text{open left closed } u_{K,p,i,j} < 0 \\ 0 & \text{right closed} \end{cases} \quad (2.38)$$

¹Note that the diagonal coefficient of matrix \mathbf{A} , $a_{K,p,i,j}$, which is always written with four dimensional indices, should not be confused with age notation a , nor with discrete age intervals a_p .

For a given $i = 1, \dots, n_x$ elements of $\mathbf{B}_{K,p,i}$ are given by

$$d_{K,p,i,j} = \begin{cases} -\frac{v_{K,p,i,j-1}}{\Delta y} - \frac{D_{K,p,i,j} + D_{K,p,i,j-1}}{2\Delta y^2} & \text{open bottom closed } v_{K,p,i,j} > 0 \\ -\frac{D_{K,p,i,j} + D_{K,p,i,j-1}}{2\Delta y^2} & \text{open bottom closed } v_{K,p,i,j} < 0 \\ 0 & \text{top closed} \end{cases} \quad (2.39)$$

$$e_{K,p,i,j} = \begin{cases} \frac{2}{\Delta t} + \frac{v_{K,p,i,j}}{\Delta y} + \frac{D_{K,p,i,j+1} + 2D_{K,p,i,j} + D_{K,p,i,j-1}}{2\Delta y^2} \\ + M_{K,p,i,j} & \text{open } v_{K,p,i,j} > 0 \\ \frac{2}{\Delta t} - \frac{v_{K,p,i,j}}{\Delta y} + \frac{D_{K,p,i,j+1} + 2D_{K,p,i,j} + D_{K,p,i,j-1}}{2\Delta y^2} \\ + M_{K,p,i,j} & \text{open } v_{K,p,i,j} < 0 \\ \frac{2}{\Delta t} + \frac{D_{K,p,i,j} + D_{K,p,i,j-1}}{2\Delta y^2} + M_{K,p,i,j} & \text{bottom closed } v_{K,p,i,j} > 0 \\ \frac{2}{\Delta t} - \frac{v_{K,p,i,j}}{\Delta y} + \frac{D_{K,p,i,j} + D_{K,p,i,j-1}}{2\Delta y^2} + M_{K,p,i,j} & \text{bottom closed } v_{K,p,i,j} < 0 \\ \frac{2}{\Delta t} + \frac{v_{K,p,i,j}}{\Delta y} + \frac{D_{K,p,i,j+1} + D_{K,p,i,j}}{2\Delta y^2} + M_{K,p,i,j} & \text{top closed } v_{K,p,i,j} > 0 \\ \frac{2}{\Delta t} + \frac{D_{K,p,i,j+1} + D_{K,p,i,j}}{2\Delta y^2} + M_{K,p,i,j} & \text{top closed } v_{K,p,i,j} < 0 \\ \frac{2}{\Delta t} + M_{K,p,i,j} & \text{closed} \end{cases} \quad (2.40)$$

$$f_{K,p,i,j} = \begin{cases} \left(-\frac{D_{K,p,i,j} + D_{K,p,i,j+1}}{2\Delta y^2} \right) & \text{open top closed } v_{K,p,i,j} > 0 \\ \left(\frac{v_{K,p,i,j+1}}{\Delta y} - \frac{D_{K,p,i,j} + D_{K,p,i,j+1}}{2\Delta y^2} \right) & \text{open top closed } v_{K,p,i,j} < 0 \\ 0 & \text{bottom closed} \end{cases} \quad (2.41)$$

2.3.2 Diagonal coefficients for Dirichlet boundary conditions

Let us provide the general formulae for movement rates as well as for diagonal coefficients of tridiagonal matrices of linear system that take into account Dirichlet boundary conditions.

First, we define the following expressions to account for the type of neighbouring grid cell

$$L_{i,j}^g = \begin{cases} 1 & \text{if left open-to-global} \\ 0 & \text{else} \end{cases} \quad (2.42)$$

$$R_{i,j}^g = \begin{cases} 1 & \text{if right open-to-global} \\ 0 & \text{else} \end{cases} \quad (2.43)$$

We define the movement rates given the closed boundary

$$\begin{aligned} D_{K,p,i-1,j} &= -D_{K,p,i,j} && \text{left-closed} \\ D_{K,p,i+1,j} &= -D_{K,p,i,j} && \text{right-closed} \\ u_{K,p,i-1,j} &= 0 && \text{left-closed} \\ u_{K,p,i+1,j} &= 0 && \text{right-closed} \\ u_{K,p,i,j} &= \max(u_{K,p,i,j}, 0) && \text{left-closed} \\ u_{K,p,i,j} &= \min(u_{K,p,i,j}, 0) && \text{right-closed} \end{aligned} \quad (2.44)$$

We update the expressions including finite differences as follows

$$\alpha_{K,p,i,j} = -\frac{\text{sign}(u_{K,p,i-1,j}) + 1}{2} \times \frac{u_{K,p,i-1,j}}{\Delta x} - \frac{D_{K,p,i,j} + D_{K,p,i-1,j}}{2\Delta x^2} \quad (2.45)$$

$$\gamma_{K,p,i,j} = \frac{1 - \text{sign}(u_{K,p,i+1,j})}{2} \times \frac{u_{K,p,i+1,j}}{\Delta x} - \frac{D_{K,p,i,j} + D_{K,p,i+1,j}}{2\Delta x^2} \quad (2.46)$$

Hence, with Dirichlet boundary conditions diagonal coefficients become

$$a_{K,p,i,j} = \alpha_{K,p,i,j} \times (1 - L_{i,j}^g) \quad (2.47)$$

$$b_{K,p,i,j} = \frac{2}{\Delta t} + \text{sign}(u_{K,p,i,j}) \frac{u_{K,p,i,j}}{\Delta x} \quad (2.48)$$

$$\begin{aligned} &+ \alpha_{K,p,i,j} \times L_{i,j}^g + \gamma_{K,p,i,j} \times R_{i,j}^g \\ &+ \frac{D_{K,p,i+1,j} + 2D_{K,p,i,j} + D_{K,p,i-1,j}}{2\Delta x^2} + M_{K,p,i,j} \end{aligned}$$

$$c_{K,p,i,j} = \gamma_{K,p,i,j} \times (1 - R_{i,j}^g) \quad (2.49)$$

Similar expressions can be obtained for the coefficients of matrix B . The same formulas for the right-hand-side vector elements as eq. (2.32) and eq. (2.33) are simply updated with Dirichlet boundary conditions diagonal elements $d_{K,p,i,j}$, $e_{K,p,i,j}$, $f_{K,p,i,j}$ and $a_{K,p,i,j}$, $b_{K,p,i,j}$, $c_{K,p,i,j}$ (2.48-2.49) for the x and y -direction respectively.

2.4 Algorithm

2.4.1 Initial conditions

Each cohort $p \geq 0$ is initialised at initial time, for all i, j as

$$N_{0,p,i,j} = \mathcal{N}_{p,i,j}, \quad p = 0 \dots n_a. \quad (2.50)$$

To obtain \mathcal{N} in order to initialise the model, we use either a previous estimation of the population density if available for a given time step (so called restart), or generate this vector designing the spin-up run (see Chapter 4 for more details).

2.4.2 Age initialisation

The density of larvae, $N(K, 0, i, j)$, that is, the population density of the age class $p = 0$, is ‘initialised’ at each time step $K = 1, \dots, n_T$ in all i, j according to eq. (1.13). This initialisation takes place after time and age integration (sections 2.2.1 and 2.2.2).

2.4.3 Main loop

The population density is initialised before entering into the time loop. The age loop is embedded into the time loop. The outer loop includes the updates of the environmental and fisheries data, the update of habitats and all dynamic rates for each age class, resolution of numerical model with the ADI method, age integration, which occurs once the full step $[T_{K-1}, T_K]$ of the discretised ADR model is done, and finally the age initialisation. These steps are executed as detailed below:

- Initialise population density as in eq. (2.50) at $K = 0$ for all $p = 0, \dots, n_a$;
- Run the outer time loop for $K = 1, \dots, n_T$:
 - update all forcing variables for time T_K ;
 - do age loop: for $p = 0, \dots, n_a$:
 - * update coefficients of discretised advection–diffusion–reaction eq. (1.1), $u_{K,p}$, $v_{K,p}$, $D_{K,p}$, $M_{K,p}$;
 - * compute the elements of all matrices $\mathbf{A}_{K,p,j}$, $j = 1, \dots, n_y$ (2.34) for the first half time steps;
 - * compute the elements of all matrix $\mathbf{B}_{K,p,i}$, $i = 1, \dots, n_x$ (2.35) for the second half time steps;
 - * do inner loop of the ADI solver for $k = 1, \dots, n_t$:

- ◊ loop for $j = 1, \dots, n_y$ to solve n_y linear systems (2.30) along the x -direction at the first half time step:
 - i) compute right-hand side elements $g_{k,j}$ (2.32) using the solution of linear system either at $k = 0$ (for $K - 1$) or at $k + 1$, that is, $\mathbf{N}_{k+1,p,i}$;
 - ii) solve system (2.30) to get $\mathbf{N}_{k+\frac{1}{2},p,j}$ vector;
- ◊ loop for $i = 1, \dots, n_x$ to solve n_x linear systems (2.30) along the y -direction at the second half time step:
 - i) compute right-hand-side elements $h_{k+\frac{1}{2},i}$ (2.33) using the solution $\mathbf{N}_{k+\frac{1}{2},p,j}$ of the first half-step;
 - ii) solve system (2.31) to get $\mathbf{N}_{k+1,p,i}$ vector, at $k = n_t$ save the model numerical solution corresponding to time step T_K ;
- do age integration as described by eqs. (2.10);
- compute the new larval density $N_{K,0}$ given by eq. (1.13);
- save the state vector and auxiliary variables for each time step T_K .

References

- Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. 1992. Numerical recipes in C. Cambridge University Press, New York.
- Sibert, J.R., Fournier, D.A. 1994. Evaluation of advection-diffusion equations for estimation of movement patterns from tag recapture data. In Interactions of Pacific tuna fisheries, FAO Summary report and papers on interaction, 1: 108-121.
- Sibert, J.R., Hampton, J., Fournier, D.A., Bills, P.J. 1999. An advection-diffusion-reaction model for the estimation of fish movement parameters from tagging data, with application to skipjack tuna (*Katsuwonus pelamis*). *Can. J. Fish. Aquat. Sci.* 56: 925-938.

Chapter 3

Configurations and model runs

3.1 System requirements

SEAPODYM is a Linux application. When it is used for simulation runs only, namely for simulating population dynamics with fixed (presumably optimal) parameters, there are no specific requirements for the computer configuration. In general, the computer power depends on the numerical model resolutions. For example, Pacific-scale simulations at 1° spatial resolution and monthly time step can be easily run on a laptop with 8GB of RAM and a 64-bit CPU operating at 2GHz. However, for higher resolutions and/or global configurations, especially when parameter estimation is envisaged, the advised minimal configuration to run the application efficiently should have a 64-bit CPU at 2GHz and higher, with at least 32 GB of RAM. The physical memory is the key requirement in the case of parameter estimation or other procedures involving the gradient computation. This is because the backward (adjoint) differentiation method stores all intermediate variables needed for the exact evaluation of a cost function gradient in the operational memory. If there is not enough RAM available, the program will dump all temporary data on the hard disk (in files cmpdiff.tmp and gradfil.tmp), which will significantly increase the overall runtime.

This chapter is for users who want to run the application with the code and executable of the program that are provided. If modifications of some part of the model code are desired, it is highly recommended that the developer's team be contacted, since changes in the code can interfere with computation of the likelihood gradient and hence alter some model functionalities.

3.2 Installations

Compilation of the source code requires installation of the two additional libraries listed below.

libxml2 library is used to read and write all application parameters in a text-based **xml** parameter file. If it is not installed already, you can download and build the library from <ftp://xmlsoft.org/libxml2>

Autodif libraries provide an array language extension to C ++ enabling the automatic code differentiation (Autodif User’s Manual, 2021). Please visit the [ADMB project](#) website for the latest release of the [ADModel Builder software](#)¹ including Autodif libraries and follow the instructions to compile and configure it.

After installation of the ADMB software, you need to declare the environment variable in the .bashrc file pointing to the address where the installed libraries can be found, for example:

```
export ADMB_HOME=/path-to-admb-folder/
```

This variable needs to be specified in the SEAPODYM Makefile. In the case when shared libraries are to be used, add the following in the .bashrc as well:

```
export LD_LIBRARY_PATH=$ADMB_HOME/lib:$LD_LIBRARY_PATH
```

Once the libraries have been installed and configured, create the source directory, place the code there, edit the provided Makefile.i64 (see Appendix B) to make sure that you have the paths and linker flags recognisable by your gcc compiler, then compile the SEAPODYM application by typing the command:

```
make -j -f Makefile.i642
```

If there are no conflicts in libraries and compiler versions, and all paths are specified correctly, the executable **seapodym_cltags** will be built in the source code directory. It is convenient to create the alias in the .bashrc, for example, **seapodym** to point to this executable, so it can be called without the full path.

3.3 Running SEAPODYM with predefined configuration

The SEAPODYM-MASS application runs in a command line. Typing the following command:

¹Note, the program will not use ADModel Builder itself, but only Autodif libraries libado and libadt.

²The option -j performs a parallel execution of make, and it can be omitted.

```
seapodym --help
```

provides the usage and the list of available running options:

Usage: seapodym [options] <parfile name>

Options:

-h, --help	Print this message and exit.
-H, --hessian	Compute Hessian matrix.
-p, --projection	Compute 2D-projection of the likelihood on a grid specified in parfile.
-s, --simulation	Run simulation without optimization.
-sa, --sensitivity-analysis[=FLAG]	Make sensitivity analysis. By default[=0] sensitivity function takes model predictions only. If FLAG=1 the sensitivity function takes both predictions and observations. If FLAG=2 ONE-AT-A-TIME sensitivity analysis. If FLAG=3 ALL-AT-A-TIME sensitivity analysis.
-t, --twin-experiment[=FLAG],	Perform identical (by default, or FLAG=0) twin experiment. If FLAG=1 the noise will be added to the artificial data.
-v, --version	Print version number and exit.

According to this prompt, the program may run with a set of different options, corresponding to available types of runs (described in section 3.5), and requires the configuration file, also called the parfile. However, prior to running the first simulation, one needs to verify the availability of all necessary input data and to configure the paths to these data in the parfile.

3.3.1 Input directories

First, place the forcing datasets that were provided together with the parfile in the input directories and organise them as shown in Figure 3.1. Create two directories, for example, run-OGCMname-domain-resolution and fisheries/spname, for the environmental and the fisheries data respectively. Provide correct absolute paths to these directories in the configuration XML file, in `strdir` and `strdir_fisheries`. Second, make sure that the environmental forcing directory contains all necessary input data in binary format and the two ASCII files called `mask *.txt` and `topo*.txt`, as shown in Figure 3.1. Note also that fisheries data should include `spname_catch_*.txt` file as well as one or several `spname_LF_*.txt` files as described in the XML file (see panels (b) and (c) in Figure 3.1).

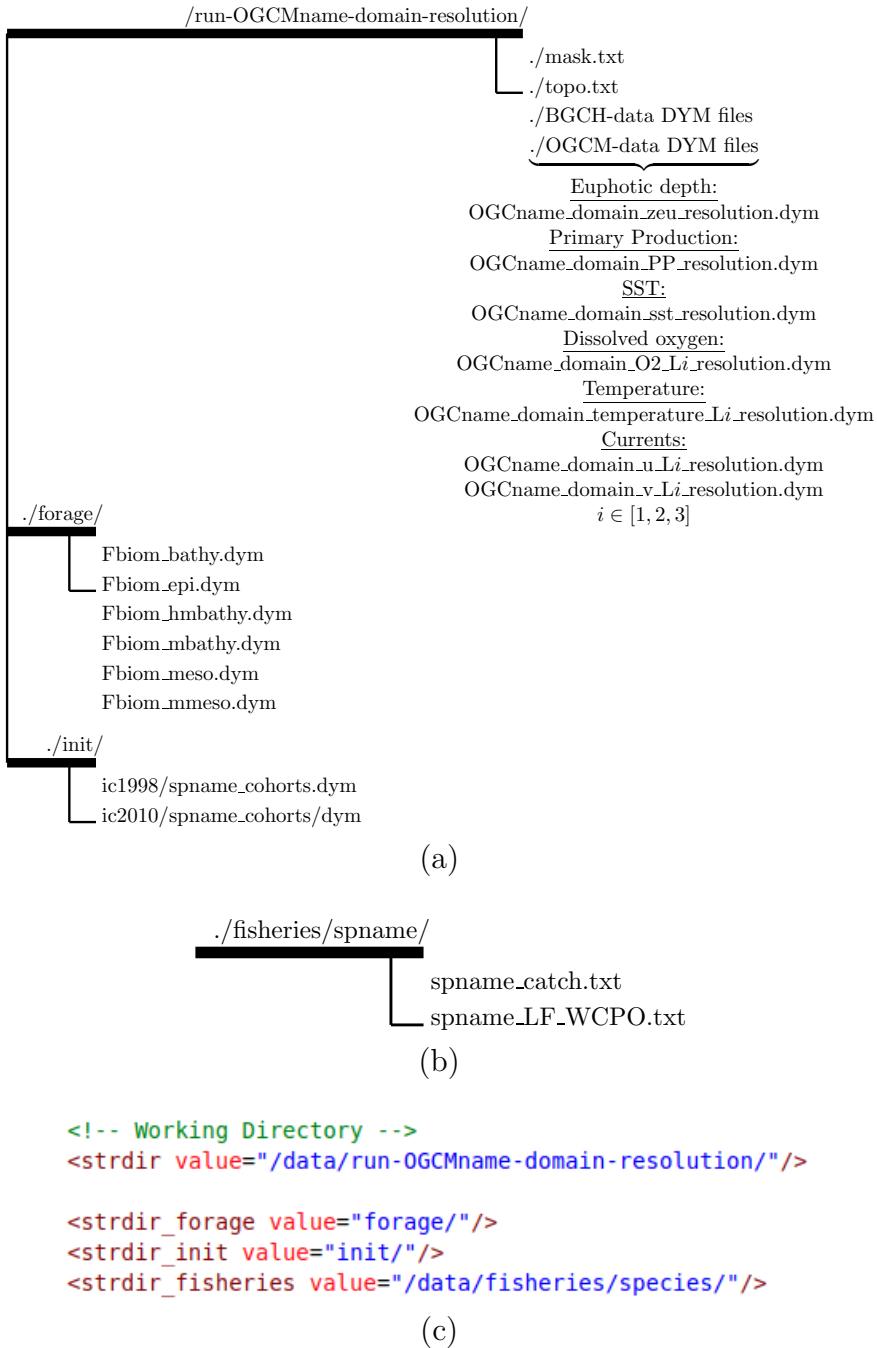


Figure 3.1: **Organising folders and files with SEAPODYM input data.** (a) The folder `run-OGCMname-domain-resolution` includes the binary DYM files with environmental data (physical, biochemical and biological forcings) as well as ASCII files with land mask(s) and auxiliary data (topographic index, Exclusive Economical Zones contours), and two subfolders with forage fields and initial conditions. (b) The fisheries data files are stored in a separate folder. (c) Input directories in the configuration file: the paths to the environmental (`strdir`) and fisheries data (`strdir_fisheries`) are absolute, whereas the paths to the initial conditions (`strdir_init`) and to the forage data (`strdir_forage`) are relative.

3.3.2 Simulation run

When everything is set, you are ready to run the first simulation. It is advisable to create a running directory, where the model will be executed and the output files will be written. The command to run a simulation is:

```
seapodym -s <parfile name>
```

Note, if the option `-s` is omitted then `seapodym` will actually start minimising the negative log-likelihood according to the configuration in the parameter file `parfile`, and only after completion of the iteration cycle will it run the simulation during which the model output will be written. If this is not desired, for example, if the reference parameter file already contains the MLE solution and the goal is to get the model outputs, then always use the simulation option.

If the first simulation ran successfully, the output folder was auto-generated in the running directory and the output files were written. In this case see section 3.6 for a detailed overview of model outputs. Otherwise, if the run ended abnormally, continue reading the following sections to understand the model configuration, to identify and troubleshoot the problem(s).

3.4 Parameter file

Configuration of a SEAPODYM run is written in an XML file, called a *parfile* (see full example of a parfile in Appendix A.2). The following sections explain model configuration through a detailed description of the parfile.

3.4.1 Spatial domain and resolution

The model spatial domain is defined by its rectangular geographic area and the complex boundaries within this area. The geographic extension is chosen in line with the knowledge of habitats and migrations of the modelled species: the domain is always set larger than the known area occupied by the population in order to avoid the impact of closed boundary conditions. The complex boundary of the computational domain is described by the land mask (see below). The spatial resolution of the two-dimensional domain is set up by two parameters, `deltaX` and `deltaY`, which define the constant step of the uniform Arakawa-A grid used to resolve model 1.1 numerically (more details in Chapter 2). Note that irregularity of the grid is handled in the code through latitudinal correction of the grid cell size. The vertical resolution is accounted implicitly in the model, and is defined by the euphotic depth that is provided as a forcing variable (see below). The number of vertical layers is fixed in the attribute `nb_layer`. This setup cannot be modified.

```

<!--=====
<!--Domain and resolution parameters-->
<!--===== -->
<latitudeMin value="-54.5"/>
<latitudeMax value="65.5"/>
<longitudeMin value="88.5"/>
<longitudeMax value="290.5"/>
<nb_layer value="3"/>
<deltaX value="120"/>
<deltaY value="120"/>

```

Figure 3.2: Example of a computational grid configuration in the parfile. Note that the longitudinal and latitudinal coordinates correspond to the corners of the uppermost grid cells and not the centres of the first and last grid cell.

An example configuration for the Pacific Ocean model domain with 2° resolution is shown in Figure 3.2. Table 3.1 provides a detailed description for each configuration parameter.

Table 3.1: Definition of the domain boundaries and the spatial resolution in the parfile. The names in the left column correspond to the names of the XML nodes.

Parameters	Description
<code>latitudeMin</code>	The latitude of the southern grid cell corner, from +90N to -90S
<code>latitudeMax</code>	The latitude of the northern grid cell corner, from +90N to -90S
<code>longitudeMin</code>	The longitude of the western grid cell corner, from 0 to 360
<code>longitudeMax</code>	The longitude of the eastern grid cell corner, from 0 to 360
<code>nb_layer</code>	The number of vertical layers (always 3)
<code>delta_x</code>	The grid resolution in longitudinal dimension, in nautical miles
<code>delta_y</code>	The grid resolution in latitudinal dimension, in nautical miles, see eq. (2.1) and Figure 2.1

3.4.2 Land mask

The land mask is a matrix with integer values of the size corresponding to the number of grid cells in longitudinal and latitudinal dimensions. The land is denoted ‘0’ and the ocean cells are defined by non-zero values as follows: 1 – only epipelagic layer exists, 2 – only two vertical layers exist (i.e., epipelagic and upper mesopelagic), and 3 – all three layers exist (epipelagic, upper and lower mesopelagic). The land mask is stored in the ASCII file and its name should be provided in the `str_file_mask` in the parfile. Note that the mask can be produced and/or modified using the *GMB* software (see Appendix Toolbox D.1).

```

<!--=====
<!--Time control      -->
<!--===== -->
<deltaT value="30"/>
<iterationNumber value="15"/>
<save_first_date year="1979" month="1"/>
<save_last_date year="1983" month="12"/>

```

Figure 3.3: Example of a set-up for the time control parameters.

3.4.3 Time control

The time control parameters include the configuration of the time stepping and the simulation time period (Table 3.2). It is not possible to modify time stepping `iterationNumber` for a given configuration, as it is determined by the temporal resolution of the forcing fields. At the same time, the user can modify the starting and the final date of the run, given that these dates belong to the interval of the forcing data availability. It is advised that the parameter `iterationNumber` is kept unchanged, but higher values can be set to improve the precision of the ADI method. An example of a setup for the time control parameters is shown in Figure 3.3. Note that the case of a 30-day time stepping (360-day calendar) is particular, as it corresponds to a virtual 360-day calendar. In this case the day can be omitted as it is automatically set to 15.

Table 3.2: Setting up the temporal resolution of the numerical method and the time range of the simulation.

Parameters	Description
<code>deltaT</code>	The time step in the outer loop of the numerical model, see ΔT in (2.4), given in days. It is usually predefined by the time step in the forcing data.
<code>iterationNumber</code>	Number of iterations in the numerical model; it determines the time step in the inner loop of the numerical model, i.e. the time step of the ADI method, see n_t in (2.5).
<code>save_first_date</code>	Starting date of run, can be any date within the time range of the forcing datasets. The user can specify the year, month and day.
<code>save_last_date</code>	Final date of run, cannot be earlier than starting date and outside the time range of the forcing datasets.

3.4.4 Forcing data

SEAPODYM relies on multiple oceanic variables to describe the environment of modelled species. These include physical, biochemical and biological forcing variables. Some

of them are three-dimensional, such as temperature, dissolved oxygen (O_2) concentration, zonal/meridional currents and primary production, while others are two-dimensional, such as euphotic depth and functional groups of micronekton. These variables are written in SEAPODYM-specific binary files .dym (Figure 3.1) that can be visualised with GUI software called SeapodymView (see Appendix D.1). The format of these DYM files is detailed in Appendix C.3.1. Note that each file contains the mask and domain information, which should correspond to the model configuration in the XML parfile.

Physical and biogeochemical forcing data

The physical forcing dataset includes three physical variables: the temperature (in $^{\circ}\text{C}$), and the zonal (u) and meridional (v) components of the currents (in $\text{m}\cdot\text{s}^{-1}$). The biogeochemical fields consist of the primary productivity (in $\text{mmol C}\cdot\text{m}^{-2}\cdot\text{day}^{-1}$), the dissolved oxygen concentration (in $\text{ml}\cdot\text{L}^{-1}$), and the euphotic depth (in m). Prior to being used by SEAPODYM, these forcing variables are always interpolated to a regular Arakawa A grid with the resolution specified by ΔX and ΔY and variable values are placed in the centre of the grid cells. Primary production used in SEAPODYM is vertically integrated throughout the water column (while mostly non-zero in the euphotic zone), whereas the other three-dimensional variables are averaged within the three pelagic layers, defined according to the euphotic depth Z_{eu} (see Figure 1.2). The three vertical layers, $z = (1, 2, 3)$, are bounded within the intervals $(0, 1.5Z_{\text{eu}})$, $(1.5Z_{\text{eu}}, 4.5Z_{\text{eu}})$, and $(4.5Z_{\text{eu}}, \min(10.5Z_{\text{eu}}, 1000\text{m}))$ (see Chapter 1, section 1.2.1 for more details). Therefore, there are nine files with physical oceanographic data, three files with dissolved oxygen data and one for the primary production.

In the parfile, besides providing the paths to the directory with forcing data (section 3.3.1 and Figure 3.1) and the names of each file (Figure 3.4), the following flags are set up in `strfile_oxy` depending on the availability of the oxygen data: 0 (default) means that the time series of oxygen with the same time step are used in the simulation, 1 means that the available oxygen data are monthly climatology, and 2 means that oxygen data have only four spatial fields, each corresponding to one calendar season (January–March, April–June, July–September, October–December).

Note that contrary to the previous SEAPODYM-MASS versions, the monthly climatology variables are no longer used in the model. For technical reasons these variables, as well as their file names, are kept in the code and can be present in the parfile; however, the program does not require these files.

Mid-trophic level biomass

The biological forcing of SEAPODYM-MASS consists of six functional groups of micronekton, representing the prey fields of modelled species. The modelling of the mid-trophic

```

<!--sea surface temperature-->
<strfile_sst value="OGCMname_domain_sst_firstyear_lastyear.dym"/>
<!--euphotic depth-->
<strfile_vld value="OGCMname_domain_zeu_firstyear_lastyear.dym"/>
<!--primary production integrated on the three layers-->
<strfile_pp value="OGCMname_eppley_vpmp_wmean_domain_pp_firstyear_lastyear.dym"/>
<!--dissolved oxygen averaged in each layer-->
<strfile_oxy layer0="OGCMname_domain_O2_L1_firstyear_lastyear.dym"
    layer1="OGCMname_domain_O2_L2_firstyear_lastyear.dym"
    layer2="OGCMname_domain_O2_L3_firstyear_lastyear.dym"/>
<!--zonal and meridional currents averaged in each layer-->
<strfile_u layer0="OGCMname_domain_u_L1_firstyear_lastyear.dym"
    layer1="OGCMname_domain_u_L2_firstyear_lastyear.dym"
    layer2="OGCMname_domain_u_L3_firstyear_lastyear.dym"/>
<strfile_v layer0="OGCMname_domain_v_L1_firstyear_lastyear.dym"
    layer1="OGCMname_domain_v_L2_firstyear_lastyear.dym"
    layer2="OGCMname_domain_v_L3_firstyear_lastyear.dym"/>
<!--temperature averaged in each layer-->
<strfile_t layer0="OGCMname_domain_temperature_L1_firstyear_lastyear.dym"
    layer1="OGCMname_domain_temperature_L2_firstyear_lastyear.dym"
    layer2="OGCMname_domain_temperature_L3_firstyear_lastyear.dym"/>
<!--type of oxygen data: 0 for time series, 1 - for monthly, 2 for seasonal-->
<type_oxy value="0"/>

```

Figure 3.4: The configuration of input forcing variables in the XML parfile using the generic names (the naming convention is not imposed) for physical and biogeochemical forcing data files.

functional groups (i.e. micronekton) is done separately by the SEAPODYM-LMTL sub-model based on the same forcing fields of temperature, currents and primary production. The six files produced by SEAPODYM-LMTL, one per functional group, contain the potential biomass (in $\text{gWW} \cdot \text{m}^{-2}$) of micronektonic organisms as shown in Figure 1.1. These variables are also written in DYM format. The names of these files are composed from prefix `Fbiom` and the names of the micronekton functional groups that are listed under the `frg_name` tag in the parfile (see section 3.4.7). Note that a non-conformance of these names in the forcing directory and the parfile will lead to the runtime error.

Initial conditions

Initial condition, that is, the model state vector at time t_0 , $N(a, t_0, x, y)$ in (eq. 1.1), is the solution of the numerical model obtained for the date preceding the beginning of the run by one time step. The file with initial conditions, also called the *restart* file (see also section 3.6.2) is automatically generated at the end of each simulation and written in the output directory. The structure of this file, written in the DYM format, is a series of two-dimensional density fields for all age classes from a_0 to $A+$ at time t_0 . Thus the third dimension, `Zlevel` (see Appendix C.3.1), in this file is age and not time. The name of this file is `spname_cohorts.dym`, where the species short name `spname` is declared in the parfile

```

<!--=====-->
<!--FISHERY DATA -->
<!--=====-->

<file_catch_data>
|   <nb_files bet="1"/>
|       <bet file1="bet_catch_16f_L3move.txt"/>
</file_catch_data>

<file_frq_data>
|   <nb_files bet="2"/>
|       <bet file1="bet_LF_14f_P0_filtered.txt"/>
|       <bet file2="bet_LF_2f_PS_EPO_normalized.txt"/>
</file_frq_data>

```

Figure 3.5: Catch and effort data files in the parfile.

under the `sp_name` tag. In order to be taken into account by the seapodym application, the initial condition file should be placed into a subfolder called `init`, and its relative path should be declared in the `strdir_init` attribute of the parfile (panel (c), Figure 3.1).

Fisheries data

The anthropogenic forcing is represented by fisheries data, that is, effort-and-catch and length frequencies. These variables are written in ASCII format (see Appendix C.1) in a tabular format and not as the two-dimensional fields. This is because the effort and catch data are usually sparse data. The path to the directory with fishing data (see Figure 3.1) is specified in `strdir_fisheries` in the parfile. As there are two types of fisheries data, there are two types of files – one file, `spname_catch_*.txt`, containing the effort and catch (EC) data and one or more files, `spname_LF_*.txt`, with length frequency (LF) data (see Figure 3.5). The formats of these files are detailed in Appendix C.1. Note, if LF data are stored in multiple files (e.g., coming from different sources with distinct regional structure), the LF data files must be listed in the parfile in the sequential order of the sampling regions found in the file headers (Appendix C.1), as file1, file2 etc. (Figure 3.5).

The temporal resolution of EC data should be the same as the resolution of the model, that is, δT . The spatial resolution of EC data can be different, but unique within a single fishery (see declaration of fisheries in section 3.4.5). The model application can handle coarser spatial resolutions of EC data. In the case when it is finer than the model spatial resolution, the data will be simply summed up within the model grid cell. Currently, the numerical model accepts only the quarterly and regional LF data.

Topographic index

One more type of input data can be used by SEAPODYM to limit the movement of fish towards coastal areas with shallow topography. The topographic index is the matrix with real

```

<!--=====-->
<!--FISHERY DATA -->
<!--=====-->

<nb_fishery value="1"/>
<list_fishery_name>L1 L2 L3 L4 L5 L6 L7 L8 S9 S10 S11 S12 S13 S14 P15 016</list_fishery_name>
<degrade_fishery_reso_deg value="5"/>

<!--Mask for fishery data usage: 0 - do not take into account-->
<mask_fishery_sp>
    <bet>1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</bet>
</mask_fishery_sp>

<mask_fishery_likelihood>
    | |
    | <bet>1 1 1 1 0 1 1 1 1 1 1 1 1 1 0</bet>
</mask_fishery_likelihood>

<!--1 - C in metric tones; 0 - C in numbers -->
<fishery_catch_units>1 1 1 1 1 1 1 1 1 1 1 1</fishery_catch_units>

<mask_fishery_no_effort>
    | |
    | <bet>0 0 0 0 1 0 0 0 0 0 1 1 0 1 1</bet>
</mask_fishery_no_effort>

<flex_regstruc value="1"/>

```

Figure 3.6: Example of the fisheries configuration in the parfile.

values between 0 and 1, which act as multipliers to the habitat index that drives movement of young and mature adult age classes (see classification in Table 1.1). The small values of this index indicate the less accessible (shallow) areas, and values ‘1’ indicate no effect of topography on the habitat and hence movement. The use of topographic index is optional. In the case when the topography is not considered, the file does not need to be specified in the parfile. Current reference configurations may include the topographic index generated with the help of the algorithm in the GMB software.

3.4.5 Fisheries configuration

There are multiple configuration parameters related to the use of fisheries data in simulations (Table 3.3). They specify the number of fisheries, their short names, the units of catch, the flags controlling the use of fisheries as the model forcing (fishing mortality), the flag for catch prediction method, the resolution and the use of each fishery in the likelihood (see example in Figure 3.6). The fishery short name, or fishery ID, is constructed from the letter corresponding to the gear code and the fishery index in file spname_catch*.txt. For example, the fishery short name L1 refers to the first fishery in the dataset being the long-line fishery.

Note that the effort and catch data provided on coarser spatial resolutions will be redistributed to the resolution of the model to compute fishing mortality, but used at the original or even coarser resolution in the catch prediction method. In other words, the predicted

catch will either be at the spatial resolution at which the original data were collected, or, if the option `degrade_fishery_reso_deg` is set to a resolution coarser than the original fisheries resolution and that of the model, then the resolution of the predicted catch of all eligible fisheries will be degraded to a specified value. The information about initial and final spatial resolutions of all fisheries will be printed in the screen log of the simulation.

Table 3.3: Fisheries configuration parameters. The names in the left column correspond to the names in the XML parameter file.

Parameters	Description
<code>nb_fishery</code>	Number of fisheries, n_f , in the current configuration.
<code>list_fishery_name</code>	List of fishery short names of length n_f . These fisheries must be present in file <code>spname_catch*.txt</code> .
<code>degrade_fishery_reso_deg</code>	Resolution (in degrees) to degrade fisheries data in the model predictions only.
<code>mask_fishery_sp</code>	A vector of integers of length n_f to activate (1) or deactivate (0) fisheries in the model run.
<code>mask_fishery_likelihood</code>	Same as above, but influences the likelihood computation only. It means that if the fishery is active in ‘ <code>mask_fishery_spname</code> ’, but turned off in this flag, it still causes fishing mortality, but predicted catch and/or LF of this fishery are not accounted in the likelihood.
<code>fishery_catch_units</code>	A vector of integers of length n_f to specify the units of the observed catch in the fishery: 1 for metric tonnes, 0 for numbers.
<code>mask_fishery_no_effort</code>	A vector of integers of length n_f to specify which fishery do not have effort data: 0 if the fishery has effort data, 1 if fishery does not have effort data. The catch removal method is used for fisheries without effort.

3.4.6 Likelihood function configuration

Regardless of the type of the run (see section 3.5), the likelihood function specified in the parfile will be computed and the current (cumulative) value will be displayed at every time step and at the end of the simulation. The negative log-likelihoods are computed and reported by data type in the following order: catch likelihood, length frequency likelihood, tagging data likelihood and the average stock function. Since this chapter is devoted to a simulation run with a predefined configuration, this section aims to provide only the short description of likelihood parameters, configured through the parfile (Table 3.4). For their meaning and respective function, please read Chapter 4.

Table 3.4: Likelihood configuration parameters. The names in the left column correspond to the names in the XML parameter file.

Parameters	Description
<code>nb_step_to_skip</code>	A number of model time steps to skip before starting augmenting the likelihood. It is advisable to skip at least one age at maturity of the modelled species in order to give the model a time to renew the population biomass based on current parameters.
<code>total_likelihood</code>	The total likelihood value corresponding to current configuration. Note, this value will be written in the <code>newparfile.xml</code> created upon finishing the model run, and not in the parfile that was provided at execution.
<code>like_c_cpue</code>	The flag for using catch or CPUE (catch per unit effort) data in the likelihood: if ‘0’ the catch is used in the likelihood, if ‘1’ the CPUE.
<code>likelihood_types</code>	A vector of integers of length n_f (Table 3.3) for likelihood type selection for each fishery: concentrated (1), log-normal (2), Poisson (3), negative binomial (4), zero-inflated negative binomial (5), truncated Poisson (6).
<code>frq_likelihood</code>	A flag to include (1) or exclude (0) the length frequency term from the likelihood.
<code>tag_likelihood</code>	A flag to include (1) or exclude (0) the tagging data term from the likelihood.
<code>tag_likelihood_only</code>	A simulation type flag. If this flag is set to ‘1’, then only the movement model of tagged tuna sub-population density will be numerically resolved. Use this flag only for tag simulator or estimation of movement parameters from tagging data (see Chapter 4).
<code>stock_likelihood</code>	A flag to add the average stock constraint into the likelihood (β in eq. 4.23): ‘1’ to include, ‘0’ otherwise.
<code>mean_stock_obs</code>	The average biomass (\bar{B} in eq. 4.23) over the rectangular region specified within this attribute. It is used only if <code>stock_likelihood</code> is activated.
<code>likelihood_parameters</code>	A set of negative binomial dispersion parameters, (β_f in eq. 4.18), one per fishery. These parameters are effective (and can be estimated) only if the <code>likelihood_types</code> are set to 4 or 5 for corresponding fisheries.
<code>prob_zero</code>	A set of negative binomial probability of null observation parameters (p_f in eq. 4.18), one per fishery. These parameters are effective (and can be estimated) only if the <code>likelihood_types</code> are set to 4 or 5 for corresponding fisheries.

3.4.7 Dynamic model parameters

Prey model parameters

Some SEAPODYM-LMTL model (Lehodey et al., 1998, 2010, 2015) parameters are used in the SEAPODYM-MASS model. Besides, SEAPODYM-MASS is designed to run in predator–prey mode, thus coupling predator and prey dynamic models. However, since all existing reference models are based on offline coupling, and the fully coupled model is not regularly maintained in the software, we do not discuss it here. Table 3.5 details all configuration parameters for micronekton. Note that those SEAPODYM-LMTL parameters that do not affect SEAPODYM-MASS solution are listed in the parfile only for information purposes.

Table 3.5: The micronekton configuration parameters in the parfile. Note that SEAPODYM-LMTL parameters marked with ‘*’ are effective only in the predator–prey coupling mode.

Parameters	Description
Tr_max*	The maximal time before recruitment to micronekton.
Tr_exp*	The slope of temperature-dependent function of recruitment time.
inv_lambda_max*	The inverse of maximal mortality rate.
inv_lambda_curv*	The slope of temperature-dependent mortality function.
c_pp	Conversion constant from primary production to wet weight of mikronekton.
E	Energy transfer coefficient from primary production to trophic level of micronekton.
nb_forage	Number of functional groups. Refers to the number of non-zero elements in matrix F (eq. 1.5).
source_frg*	The coefficients of repartition of energy transfer among functional groups of micronekton. The sum of six coefficients should always give 1.
frg_name	The names of the functional groups of preys (see Table 3.7).
day_layer	The index of the daytime vertical layer (0,1,2) of functional group (see Figure 1.1).
night_layer	The index of the night-time vertical layer (0,1,2) of functional group (see Figure 1.1).
sigma_fcte	Diffusion rate ($\text{nmi}^2 \cdot \text{day}^{-1}$).

All six functional groups of micronekton defined in the SEAPODYM-LMTL model should be declared in the parfile. Attention should be paid to the forage names (`frg_name`). First, as pointed out earlier, these names are used in the input files with micronekton densities (section 3.4.4 and Figure 3.1). Second, they provide the names of model parameters E_{ij}

(eq. 1.21) in the parfile, see Table 3.10. It is advisable not to modify these names to ensure stability of the model software. The correspondence between forage names and model notations for the micronekton functional groups is shown in Table 3.6.

Table 3.6: The forage names, the model notations and vertical layer set-up in the parfile.

frg_name	epi	meso	mmeso	bathy	mbathy	hmbathy
model notation	F_{11}	F_{22}	F_{21}	F_{33}	F_{32}	F_{31}
day_layer	0	1	1	2	2	2
night_layer	0	1	0	2	1	0

The following micronekton parameters (Table 3.5) affect the numerical solution of the SEAPODY-MASS model. The pair `c_pp` and `E` are used to compute the wet weight of plankton (Λ in eq. 1.8). The diffusion of larvae and small juveniles, which are assumed to have similar properties to the micronekton organisms, is governed by diffusion rate `sigma_fcte` defined in the micronekton model to account for small-scale movements of fish and diffusion of water masses. Most important is the definition of the vertical structure of the micronekton functional groups, described in the attributes named `day_layer` and `night_layer` (Table 3.6). These parameters are essential, as they determine key dynamic processes of the SEAPODY model, such as movements (via accessibility of predators to prey, habitats and movement rates) and larval survival. The set-up of these vectors yields the day and night structure shown in Table 3.7).

Table 3.7: Total biomass density of functional groups of micronekton present during the day/night in each vertical layer are sums of the rows/columns in matrix \mathbf{F} (eq. 1.5).

layer name	day	night
epipelagic	F_{11}	$F_{31} + F_{21} + F_{11}$
upper mesopelagic	$F_{21} + F_{22}$	$F_{32} + F_{22}$
lower mesopelagic	$F_{31} + F_{32} + F_{33}$	F_{33}

Predator model parameters

The current version of the model software supports only a single-species model, so the value of the XML tag `nb_species` should always be set to 1. The species short name, `spname`, usually corresponds to the species code of the [FAO](#) convention. Note, the species short

```

<!--=====
<!--PREDATOR MODEL AGE STRUCTURE-->
<!--===== -->
<nb_species value="1"/>

<sp_name>bet</sp_name>

<nb_life_stages bet="3"/>
<life_stage>
|   <bet>larvae juvenile adult</bet>
</life_stage>

<nb_cohort_life_stage>
|   <bet>1 2 82</bet>
</nb_cohort_life_stage>

```

Figure 3.7: The example of the species declaration and configuration of the life stage structure for bigeye tuna in the parfile.

name provided in `nb_species` is used throughout the parfile as well as in various model outputs (file names, headers etc., see section 3.6). For example, the short name ‘bet’ is used for bigeye tuna (Figure 3.7), and all XML attributes describing species parameters, either related to its fisheries, to the likelihoods or to the model, have this short name.¹ Bigeye model parameters then appear with the following formatting:

```
<parameter_name bet="value"/>
```

See other examples in Figure 3.7 and the parfile in Appendix A.2.

Age and life stage parameters

Age and life stage parameters are essential configuration parameters that describe the species’ age and life stage structure as well as its main biological characteristics – maturity, length and weight at age (Table 3.8). All these parameters are used as predefined constants in the model, meaning that they do not vary in time and space and currently cannot be estimated. However, since the age structure plays an important role in the definition of habitats, movement rates, recruitment, mortality and predicted fisheries statistics (eqs. 1.12, 1.19, 1.20, 1.38, 4.2), modification of these parameters can alter model outputs and provide the model state, which does not correspond to the MLE solution. Therefore it is advisable not to modify these parameters in the reference configuration without subsequent parameter estimation (see Chapter 4).

¹Note, in the case of a short name modification, simply replace all occurrences in the XML file with the new name.

Table 3.8: Configuration of the species life stages, age structure, growth and maturity parameters.

Parameters	Description
<code>nb_life_stages</code>	The number of the species life stages to be considered in the model (see Table 1.1 for more details).
<code>life_stage</code>	List of names of the life stages (Figure 3.7).
<code>nb_cohort_life_stage</code>	A vector of size <code>nb_life_stages</code> giving the number of age classes by life stage.
<code>sp_unit_cohort</code>	A vector of Δa (the size of age class in days) of length $n_a = \sum \text{nb_cohorts_life_stage}$.
<code>age_recruit</code>	Age at recruitment to the exploited stock. It is used solely for comparison with stock assessment models.
<code>age_mature</code>	The age at first maturity (50% maturity). This parameter is ignored if values of maturity-at-age are provided explicitly for all age classes.
<code>maturity_age</code>	The vector of maturity-at-age estimates ($\mu(a)$ in eq. 1.12) of length n_a . The values between 0 and 1 give the proportion of mature adults in each age class.
<code>length</code>	A vector of mean fork lengths of fish ($\ell(a)$ in eq. 1.14, in cm) of length n_a , estimated in the middle of each age class.
<code>weight</code>	A vector of mean weights of fish (w_a in eq. 1.15, in kg) of length n_a , estimated in the middle of each age class.

There are a few rules to know with respect to age structure definitions in SEAPODYM. First, the number of age classes in the larvae cohorts is always 1, whatever the size of the age classes, Δa . Second, all age classes but the last one, $A+$, have the same size. It is best to have $\Delta a = \Delta T$; however, the software can manage the cases when $\Delta a > \Delta T$, if Δa is divisible by ΔT . Note that in the latter case, the ageing dynamics will be computed every $\frac{\Delta a}{\Delta T}$ time steps (see section 2.2.1 in Chapter 2); hence the model's numerical solution can be considered valid only at the end of Δa intervals. Third, the young (immature adult) life stage does not need to be specified in `life_stage` and `nb_cohort_life_stage` as it is defined automatically by the maturity vector `maturity_age` or the age at first maturity `age_mature`.

Note, the section devoted to the description of the species age structure may contain one more vector, called `age_compute_habitat`, which is not related to the definition of the species age structure, but can be specified to shorten the runtime of the simulation by avoiding re-computing habitat values (and hence velocities) until significant change in mean length occurs between age classes. This can be especially useful in the optimization runs (see section 4.4).

MLE parameters

This section describes parameters that can be estimated via the Maximum Likelihood Estimation method. All the dynamic model parameters that can be set as control parameters in optimization have the following format:

```
<parameter spname = "value">
<variable min="minimal_value" max="maximal_value" use="true"/>
</parameter>
```

where the ‘value’ is the current parameter value in the model, and the minimal and maximal values are the boundaries within which the parameter is allowed to vary in optimization. The attribute `use="true"` activates the use of a given parameter as a control variable in optimization. Any other word put in the attribute ‘use’ removes the parameter from the list of control variables. Note that boundary values are designed for the optimization runs, but they can affect the parameter value in the simulation run as well if (and only if) the attribute `use="true"` and the value of a given model parameter lies outside its boundaries. In this case the software automatically resets the parameter to a value close to the nearest boundary.

Tables 3.9–3.12 summarise model parameters that can be estimated. These tables are extensions of Table 1.3 with SEAPODYM dynamic model parameters, providing also the parfile names¹ in order to show their correspondence with the mathematical notations in model 1.1, and linking their set-up (when necessary) with other configuration parameters in the XML file.

Table 3.9: Species demographic parameters with their notations in the parfile and in model 1.1, definitions and references.

Parfile name	θ (Eq.)	Description
<i>Recruitment</i>		
<code>nb_recruitment</code>	r (1.13)	Reproduction rate in Beverton–Holt function, in mo^{-1} .
<code>a_adult_spawning</code>	b (1.13)	Slope parameter in Beverton–Holt function, in Nb/km^2 .
<i>Natural mortality</i>		
<code>Mp_mean_max</code>	\bar{m}_p (1.36)	Predation mortality rate at age 0, in mo^{-1} .
<code>Mp_mean_exp</code>	β_p (1.36)	Slope coefficient in predation mortality.
<code>Ms_mean_max</code>	\bar{m}_s (1.37)	Senescence mortality rate at age 0, in $\text{mo}^{-1-\beta_s}$.
<code>Ms_mean_slope</code>	β_s (1.37)	Slope coefficient in senescence mortality.
<code>M_mean_range</code>	ϵ (1.40)	Variability of mortality rate with habitat index from $\frac{M}{(1+\epsilon)}$ in the worst habitat to $M(1 + \epsilon)$ in the best habitat.

¹Note also, that the names in the parfile are the same as in the SEAPODYM computer code.

Table 3.9 includes demographic, that is, reproduction and mortality parameters. Note that there are other model parameters that affect the population size and age composition, such as spawning habitat (see eq. 1.13) and fisheries parameters (eq. 1.42), but compared to demographic parameters, the latter have spatial effects and/or additional functions. For example, spawning habitat parameters can modify the total number of larvae recruited as well as overall population spatial distribution. Fisheries parameters affecting the total population mortality also control the model predictions, catch and length frequency.

Table 3.10 describes all habitat-related parameters, for both spawning and feeding habitats (see sections 1.2.2 and 1.2.5). There exist two alternatives for the use of spawning habitat parameters. First, all habitat parameters can be used independently exactly as presented in Table 3.10. Second, the assumption that the species' preferred range and the target temperature are the same as the range and the temperature that are optimal for larval survival (see Chapter 1, section 1.2.5), can be set up via the XML parameter `uncouple_sst_larvae spname="0"` (see also section 3.4.7 below). Thus, assuming that $\sigma_0 = \sigma$ and $T_0 = T^*$, this set-up reduces the number of model parameters. In this case, parameters `a_sst_larvae` and `b_sst_larvae` are ignored and `a_sst_spawning` and `b_sst_spawning` are used in the equation of spawning habitat index (eq. 1.7) and in the age-dependence of predators' thermal accessibility to prey organisms (eqs. 1.19 and 1.20).

Table 3.10: Habitat parameters with their notations in the parfile and in model 1.1, definitions and references.

Parfile name	θ (Eq.)	Description
<i>Spawning habitat index</i>		
<code>a_sst_larvae</code>	σ (1.7)	Standard deviation in temperature Gaussian function of spawning habitat, when <code>uncouple_sst_larvae spname="1"</code> , in °C.
<code>b_sst_larvae</code>	T^* (1.7)	Optimal water temperature for larvae survival, when <code>uncouple_sst_larvae spname="1"</code> , in °C.
<code>alpha_hsp_prey</code>	α (1.8)	Prey encounter rate in Holling type III function, in day ⁻¹ .
<code>alpha_hsp_predator</code>	α_F (1.9)	Log-normal mean parameter in predator-dependent function, in g/m ² .
<code>beta_hsp_predator</code>	β_F (1.9)	Log-normal shape parameter in predator-dependent function.
<i>Thermal accessibility and age-dependence</i>		
<code>a_sst_spawning</code>	σ_0 (1.20)	Standard deviation in temperature Gaussian function for age 0, when <code>uncouple_sst_larvae=1</code> , otherwise it becomes σ in spawning habitat, in °C.
<code>a_sst_habitat</code>	σ_m (1.20)	Standard deviation in temperature Gaussian function at age A+, in °C.

Table continued

Parfile name	θ (Eq.)	Description
b_sst_spawning	T_0 (1.19)	Preferred temperature for age 0, when uncouple_sst_larvae=1, otherwise it becomes T^* in spawning habitat, in °C
b_sst_habitat	T_m (1.19)	Preferred temperature for the oldest adult class A+, in °C
T_age_size_slope	b_T (1.19)	Allometric power coefficient for thermal preferences at age.
<i>Oxygen limitation of the accessibility to prey</i>		
a_oxy_habitat	γ (1.18)	Slope in the oxygen accessibility function.
b_oxy_habitat	\hat{O} (1.18)	Minimal threshold oxygen value, required by the predator to access the habitat for foraging, in ml/L.
<i>Micronekton selection parameters</i>		
eF_habitat_ephi	E_{11} (1.21)	Contribution of epipelagic forage to the feeding habitat index.
eF_habitat_meso	E_{22} (1.21)	Contribution of mesopelagic forage to the feeding habitat index.
eF_habitat_mmeso	E_{21} (1.21)	Contribution of migrant mesopelagic forage to the feeding habitat index.
eF_habitat_bathy	E_{33} (1.21)	Contribution of lower mesopelagic forage to the feeding habitat index.
eF_habitat_mbathy	E_{32} (1.21)	Contribution of migrant lower mesopelagic forage to the feeding habitat index.
eF_habitat_hmbathy	E_{31} (1.21)	Contribution of highly migrant lower mesopelagic forage to the feeding habitat index.

Table 3.11 summarises the parameters used to compute species movement rates. It includes both large-scale seasonal migration parameters, and the parameters of small-scale movements governed by either spawning or feeding habitat index (see [Active random movement](#) and [Seasonal spawning migrations](#) in section 1.2.5). The seasonal migrations mechanism is optional. It can be activated (deactivated) by selecting 1 (0) in `seasonal_migrations` node. This option is usually set for sub-tropical species such as albacore and bluefin tuna, which show seasonal spawning dynamics. For tropical tunas such as skipjack, yellowfin and bigeye, which spawn opportunistically throughout the year once physiological conditions are met, this option is deactivated. In this case parameters `spawning_season_start` and `spawning_season_peak` do not affect the model solution.

Finally, the last group of parameters that are subject to estimation from observational data consists of fisheries parameters. Table 3.12 shows the largest set of parameter that can be used to define fishing mortality and to predict model catch and length frequency for a

single fishery. In practice, only part of these parameters is used given the catch prediction method and the type of selectivity function (read sections 1.2.7 and 4.1.1). The choice of selectivity function s_f (eq. 1.43) is done by fishery via `function_type` attribute within the corresponding `s_sp_fishery` node (see example of parfile fisheries setup in Appendix A.2). Setting its value to ‘1’ gives the logistic type of selectivity function (type I), value ‘2’ sets the sigmoid function (type II) and value ‘3’ sets the asymmetric Gaussian function (type III).

Table 3.11: Movement rates and parameters controlling seasonal migrations with their notations in the parfile and in model 1.1, definitions and references.

Parfile name	θ (Eq.)	Description
<i>Adult seasonal spawning migrations</i>		
<code>spawning_season_peak</code>	\hat{t} (1.33)	Mid-date (day of the year) of seasonal spawning migrations of adults.
<code>spawning_season_start</code>	ϱ_s (1.34)	Critical value of day–night ratio (or day length gradient), ϱ , triggering seasonal migrations. It also controls the duration of spawning season at each latitude.
<i>Movement</i>		
<code>MSS_size_slope</code>	A (1.29)	Slope coefficient in allometric function for tuna velocity.
<code>MSS_species</code>	V_m (1.29)	Velocity at maximal habitat gradient and $A = 1$, BL/s.
<code>sigma_species</code>	σ (1.30)	Multiplier for the theoretical diffusion rate $\frac{\bar{V}^2 \Delta T}{4}$.
<code>c_diff_fish</code>	c (1.30)	Coefficient of diffusion variability with habitat index.

Table 3.12: Fisheries parameters with their notations in the parfile and in model 1.1, definitions and references.

Parfile name	θ (Eq.)	Description
<i>Fishing mortality</i>		
<code>q_sp_fishery</code>	q_f (1.42)	Constant catchability coefficients for fishery f .
<code>s_sp_fishery</code>	ς_f (1.43)	Steepness of selectivity function if type I or II. Standard deviation if type III is set up.
<code>length_threshold</code>	\hat{l}_f (1.43)	Threshold length if selectivity function of type I. Mean length if type III selectivity.
<code>right_asymptote</code>	μ_f (1.43)	Asymptotic value of selectivity in type III function.

Dynamic model options

Here we summarise several options available in the SEAPODYM software to activate different model mechanisms and/or functional relationships. All these mechanisms and functions are explained in Chapter 1. Options are set up in the parfile with the help of integer flags. Their actions are detailed in Table 3.13. See also section 3.4.7, where the options `uncouple_sst_larvae` and `seasonal_migrations` are described.

Table 3.13: Available options in the predator dynamic model.

Parameters	Description
<code>seasonal_migrations</code>	Set to ‘1’ if the species does seasonal migrations, ‘0’ otherwise. (see section 1.2.5)
<code>vertical_movement</code>	Set to ‘1’ to account for the horizontal velocity correction due to vertical migrations (see section 1.2.5).
<code>cannibalism_juv</code>	Leave at ‘0’, not recommended due to weak observability of early life stage dynamics (see section 1.2.6).
<code>uncouple_sst_larvae</code>	Set to ‘1’ to remove the link between thermal habitat parameters for spawning habitat index and those for adult habitat-at-age (see eqs. (1.7), (1.19) and (1.20)).
<code>gaussian_thermal_function</code>	Set ‘1’ to select a Gaussian function (see (1.17)), otherwise ‘hat’ function for temperature in accessibility to prey. Currently not used.
<code>food_requirement_in_mortality</code>	Activates non-linear density dependent effect in adult mortality, accounting for intra-specific competition. Currently not used. It is recommended this be kept at ‘0’ (see section 1.2.6).

Aggregation zones

The aggregation zones can be defined for regional extractions of biomass and catch statistics, mainly for comparison with Multifan-CL model stock assessment. The extractions over EEZ (Exclusive Economic Zones) polygons can also be useful in model analyses and validations. Table 3.14 describes the configuration parameters setting up the regional structure, both for rectangular regions and EEZs. If `nb_region` is non-zero then the SEAPODYM output file SumQArea.txt will contain the biomass by life stage and catch statistics by fishery, extracted over defined regions (see more details on the SumQArea.txt file in section 3.6.3). Otherwise, this file will be empty. Likewise, if the number of EEZ areas `nb_EEZ` is non-zero and the EEZ mask file is provided (Table 3.14), then the output files, one by EEZ, called SumEEZ_eezname.txt will be written in the output directory (see section 3.6.3). The mask

file with EEZ polygons can be generated by the GMB software (Appendix D.2).

The regional structure is usually set up to match the Multifan-CL regions. The attributes named `areaN` with $N = 0, \dots, (\text{nb_region} - 1)$ contain the corners of rectangular regions (see example on panel (a), Figure 3.8). The parameters `nb_region_sp_B` and `area_sp_B` allow the selection of the subset of defined regions. It can be practical to compare the biomass predictions only in areas observed through fisheries data, which define estimations of a stock assessment model. If the configuration parameter `use_mask_catch` is set to 1, then the SEAPODYM biomass will be extracted only from the grid cells with total catch (computed over the simulation time period) exceeding 1% of the total catch over the domain.

The extractions over EEZ areas can be done based on the EEZ mask file, which has the same format as the landmask file (see section 3.4.2), with the mask being a matrix of dimensions of model spatial resolution $n_x \times n_y$ containing negative values in the grid cells inside EEZ or high seas areas. These values should be entered as EEZ IDs to activate EEZ extractions. Note, the EEZ mask file must contain all values set up in EEZ node. Example of EEZ extraction set-up is shown on panel (b) of Figure 3.8.

Table 3.14: Set-up parameters for the regional extractions over rectangular or EEZ areas.

Parameters	Description
<i>Stock assessment regions</i>	
<code>nb_region</code>	Number of rectangular regions.
<code>use_mask_catch</code>	A flag to activate (1) or deactivate (0) the use of catch mask in biomass extractions. If ‘1’, then only the cells with $C_{i,j}^{tot} > 1\%$ of C^{tot} will be used in biomass extraction.
<code>areaN</code>	The IDs and coordinates of $0, \dots, (\text{nb_region} - 1)$ rectangular region.
<code>nb_region_sp_B</code>	The number of regions selected for modelled species. In single species simulations <code>nb_region_sp_B</code> is usually the same as <code>nb_region</code> .
<code>area_sp_B</code>	Vector of indices of considered regions with values from ‘1’ to <code>nb_region</code> .
<i>Exclusive Economic Zones</i>	
<code>nb_EEZ</code>	Number of EEZ areas to be used for biomass extractions.
<code>str_file_maskEEZ</code>	EEZ mask file with negative values in grid cells inside EEZs or high seas area.
<code>eezN</code>	Names and IDs of $0, \dots, (\text{nb_eez} - 1)$ EEZ areas provided in the <code>str_file_maskEEZ</code> file.

```

<!-- ===== -->
<!-- AGGREGATION ZONES -->
<!-- ===== -->
<nb_region value="11"/>
<use_mask_catch value="0"/>
<area0 area_id="1" lgmin="120" lgmax="170" ltmin="20" ltmax="50"/>
...
<area10 area_id="11" lgmin="140" lgmax="170" ltmin="10" ltmax="20"/>

<nb_region_sp_B bet="11"/>
<area_sp_B>
|   <bet>1 2 3 4 5 6 7 8 9 10 11</bet>
</area_sp_B>

(a)

<nb_EEZ value="0"/>
<str_file_maskEEZ value="SP_IPSL_mask_eez_wcpfc.txt"/>
<EEZ>
    <eez0 name="New Caledonia" id="-19"/>
    <eez1 name="Fiji" id="-12"/>
    <eez2 name="Vanuatu" id="-33"/>
    <eez3 name="Solomon Islands" id="-29"/>
    <eez4 name="New Zealand" id="-23"/>
    <eez5 name="Norfolk" id="-20"/>
    <eez6 name="Matthew Hunter" id="-16"/>
</EEZ>

(b)

```

Figure 3.8: Configuration of aggregation zones in the bigeye model parfile. (a): Rectangular regions for comparisons with stock assessment models. (b): EEZ areas configuration. Note that by convention the EEZ IDs are always negative numbers.

3.5 Running modes

The SEAPODYM computer application is not limited to a simulation mode executing numerical resolution of model 1.1. It is the modelling framework that allows running simulations, estimating model parameters, generating and estimating parameters of habitat indices alone, running tag simulator or a virgin population model with observed density used to inform model parameters. There are also additional functionalities necessary in the development of quantitative models. They include sensitivity analyses, Hessian matrix evaluation and likelihood profiling. This section explains how to set up available running modes, to run the application in these modes, and what to expect during the runtime.

3.5.1 Simulation

The basics of a simulation run are explained in section 3.3.1. At the end of the simulation run, the parfile named `newparfile.xml` is written in the run directory and the restart file with final distributions of modelled density at age (section ??) is saved in the output directory. Note that during a simulation run all parameter values are fixed regardless of the declaration in the `use` attribute. The exception occurs if the value of a control parameter (`use="true"`) is outside its boundaries, in which case the parameter value is reset to the value close to the nearest boundary before the numerical model resolution. The warning appears in the screen log. This behaviour is coded primarily for the optimization run, but is conserved for all other types of run.

3.5.2 Habitats

This is another type of a simulation run designed to compute habitats only, either spawning or feeding for a given age(s). It is executed with a standalone model build as a result of compilation of SEAPODYM source code with the `Makefile_habitats`, providing the executable binary file called `seapodym_habitats`. This binary expects the following configuration parameters to be present in the parfile:

```
<habitat_run_type="flag" nb_ages="value"/>
<habitat_run_ages> index1 index2 ... indexN </habitat_run_ages>
```

where the `flag` ‘0’ activates the spawning habitat simulation and `flag=1` activates the feeding habitat simulation. Since the spawning habitat does not depend on age, only this flag is taken into account. In addition, the following DYM file accessible from the run directory, in sub-folder ‘habitats’ should be provided for the spawning habitat simulation:

```
<spname>_spawning_habitat_input.dym.
```

The time series of spatial distributions in this DYM file are assumed to contain habitat observations, which will be used by the MLE method in the estimation of spawning habitat parameters. The temporal dimension and the time interval of the time series in the input DYM file is required to match the run configuration. Note, if the purpose of this run is simulation only, the input DYM file does not affect the model outputs. In this case, a ‘false’ input, for example, a softlink to any DYM file with a SEAPODYM variable, can be used.

For the feeding habitat simulation, set the `habitat_run_type` to ‘1’ and choose the `value` to indicate the number of age classes for which the feeding habitats should be simulated. The attribute `habitat_run_ages` allows the age classes to be selected, with the indices corresponding to those of the age classes defined in the parfile (note, count starts at 0). Likewise, for the feeding habitat simulations, the input DYM files need to be provided as well in the run directory, sub-folder ‘habitats’. However, in the case of an age-dependent

feeding habitat, the software will require one file per age class:

```
<spname>_feeding_habitat_input_age1.dym
<spname>_feeding_habitat_input_age2.dym
...
<spname>_feeding_habitat_input_ageN.dym.
```

It is the user's responsibility to make sure that the input files contain the data for selected age classes. Then the habitat simulation is run by the command:

```
seapodym_habitats -s parfile.xml.
```

and the estimation of habitat parameters is launched by the command:

```
seapodym_habitats parfile.xml
```

See more details on the optimization mode below and in Chapter 4.

3.5.3 Density

Just as for the habitat simulation runs, this running mode can be executed by the binary file compiled with a different Makefile, called **Makefile_densities** and the input data is required for estimating the parameters based on fitting to the observed (or modelled) density fields. The input file should contain the aggregated over age dimension density, called **spname_density_input.dym**. The simulation run starts by the following command:

```
seapodym_densities -s parfile.xml,
```

while the parameter estimation run is activated without any option.

3.5.4 Optimization

As pointed out above, the SEAPODYM application launched without any option will start the run with optimization. A prompt will appear waiting for a confirmation that the intention was to run the optimization method. However, the user is strongly advised to read Chapter 4 prior to the use of this running mode. Several caveats related to the use of memory can lead to the execution problems in the configurations with high resolutions and large model dimensions, in which case the optimization run may require an increase in volume for memory allocation.

If executed successfully, at the end of the optimization experiment the program runs a simulation run with estimated parameters. The `newparfile.xml`, written at the end of this simulation will contain the updated values of the total likelihood function and model parameters. More details on the execution of optimization runs can be found in Chapter 4, section 4.4.

3.5.5 Likelihood profiling

This running mode executes the likelihood profiling with respect to two selected parameters. This mode is activated with the command:

`seapodym -p parfile.xml` with the following (example) configuration parameters to be added in the XML parfile:

```
<hyperspace_projection>
  <variables nb="2"/>
  <spname name="a_sst_spawning" nsteps="5" />
  <spname name="nb_recruitment" nsteps="5" />
</hyperspace_projection>.
```

In this example, parameters `a_sst_spawnin` and `nb_recruitment` were chosen from the list of control parameters. As a result, the program will split the parameter plane bounded by values specified by `min` and `max` attributes (section 3.4.7) into 25 intervals and run simulations to evaluate the likelihood within each interval. During the run, SEAPODYM writes an ASCII file in a matrix format with a header, called `hyperproj.out`, of the following format:

```
v1.name v2.name
nbp1 nbp2
v1.val1 v1.val2 ... v1.val.nbp1
v2.val1 v2.val2 ... v2.val.nbp2

<nbp1 x nbp2 matrix with likelihood values>
```

where `v1` and `v2` denote two control variables (model parameters) and `nbp1` and `nbp2` are the number of points between (`min`, `max`) boundaries of each parameter.

3.5.6 Hessian computation

The Hessian is computed in order to evaluate the uncertainty of estimated model parameters. This mode is practical only after the convergence of optimization to the set of MLE parameters. For the sake of time saving, the computation of the Hessian is not done automatically after each optimization run, and is launched manually, once the minimum is attained, and written to the `newparfile.xml`

```
seapodym -h newparfile.xml
```

The ASCII file generated by this run, called `Hessian.out`, contains the following information: number of variable parameters, their names, values, likelihood gradient components with respect to each variable, Hessian determinant, eigenvalues and variance-covariance matrix, organised as follows:

```
nvar
names values derivatives
det(H)
eigenvalues
variance-covariance matrix.
```

3.5.7 Sensitivity analyses

Two types of sensitivity analyses are implemented in the SEAPODYM modelling framework. These are: i) local sensitivity analysis based on gradient information, computing either the likelihood function gradient or the gradient of a quadratic function including model predictions only; ii) global sensitivity analysis including evaluation of the first-order sensitivity indices and the total-effect sensitivity indices (see Chapter 4, section 4.2 for more details). These analyses can be activated simply by running the `seapodym` application with options `-sa=FLAG`, where FLAG takes values between 0 and 4 to execute one of the following runs:

- 0: local sensitivity analysis based on evaluation of the gradient of model prediction function; each run generates a single vector of local sensitivity measures, with number of elements being the number of parameters;
- 1: local sensitivity analysis based on evaluation of the gradient of the likelihood function, thus including both predictions and observations; same as for flag ‘0’ - sensitivity measures are evaluated for a given parameter value;
- 3: a series of jitter simulation runs as a part of One-At-a-Time type of global sensitivity analysis; each run will evaluate the function value multiple times by varying only one parameter while keeping others fixed;

- 4: a simulation run as a part of All-At-a-Time type of global sensitivity analysis; the outcome of this simulation is simply the output function (usually selected likelihood term) value.

3.6 SEAPODYM Outputs

A simulation run produces multiple outputs: a series of text ASCII files aggregating predicted variables by region (see 3.4.7) or all domain, and binary DYM files with 3D (2D space and time/age) variables. This section provides information on the contents of these outputs. The formats of ASCII and DYM files are described in Appendix C, sections C.2.1–C.3. The GUI tool SeapodymView that can be used for manipulation and visualisation of SEAPODYM input and output files, both binary DYM files and ASCII files with fisheries data, is presented in Appendix D.

3.6.1 Setting the outputs directory

The path to the output directory is provided in the parfile as follows:

```
<strdir_output value = "output/">
```

where the directory called `output` is the default location of model outputs, which is created (if not existing) in the run directory. The full path to an existing directory can also be specified. Note, if the specified directory does not exist, the program will write outputs to the default location.

3.6.2 Outputs in DYM files

The two-dimensional variables are written in SEAPODYM in binary DYM format. Table 3.15 provides a complete list of files, units and writing rules.

Table 3.15: All DYM files generated by a simulation, either by default or at user's request.

Name	Description	Units	Status
Population density			
<code>spname_larve</code>	density of the larvae in number of individuals (Nb) per square kilometer	$\text{Nb} \cdot \text{km}^{-2}$	default
<code>spname_juvnl</code>	density of juveniles (sum of density from age 1 to the last specified age class in juvenile stage)	$\text{Nb} \cdot \text{km}^{-2}$	default
<code>spname_recru</code>	density at age of recruitment to the exploited population	$\text{Nb} \cdot \text{km}^{-2}$	default

Table continued

Name	Description	Units	Status
<code>spname_young</code>	biomass density of young (sum of density across all age classes within young life stage) in metric tonnes (mt) per square kilometer	$\text{mt} \cdot \text{km}^{-2}$	default
<code>spname_adult</code>	biomass density of adults (sum of density across the adult life stage)	$\text{mt} \cdot \text{km}^{-2}$	default
<code>spname_totbm</code>	total biomass density (sum of young and adult life stage)	$\text{mt} \cdot \text{km}^{-2}$	default
<code>spname_ageN</code>	density of age class N. There should be a line in the parfile <code>write_all_cohorts_dym value="1"</code>	$\text{Nb} \cdot \text{km}^{-2}$	on demand
Fisheries data			
<code>spname_Cobs</code>	total observed catch in model units	mt or Nb	default
<code>spname_Cobs_f</code>	observed catch by fishery 'f'. There should be a line in the parfile <code>write_all_fisheries_dym value="1"</code>	mt or Nb	on demand
<code>spname_Cpred</code>	total predicted catch in model units	mt or Nb	default
<code>spname_Cpred_f</code>	predicted catch by fishery 'f'. There should be a line in the parfile <code>write_all_fisheries_dym value="1"</code>	mt or Nb	on demand
Movement rates			
<code>spname_Vtot_x</code>	zonal velocity of the oldest fish (A+ class) in nautical miles (nmi) per month (mo), including both passive (zonal current velocity) and active (taxis) components. Positive to the east.	$\text{nmi} \cdot \text{mo}^{-1}$	default
<code>spname_Vtot_y</code>	meridional velocity of the oldest fish (A+ class) including both passive (meridional current velocity) and active (taxis) components. Positive to the north	$\text{nmi} \cdot \text{mo}^{-1}$	default
<code>spname_speed</code>	speed of fish in A+ class	$\text{nmi} \cdot \text{mo}^{-1}$	default
<code>spname_diffusion</code>	diffusion rate of the oldest fish	$\text{nmi}^2 \cdot \text{mo}^{-1}$	default
Habitat indices			
<code>spname_Ha_first_maturity</code>	feeding habitat index of fish in the first 50% maturity age class in adult life stage		default
<code>spname_Ha_oldest</code>	feeding habitat index of fish in A+ age class		default
Restart file			
<code>spname_cohorts_yyyy_mm</code>	density of all age classes at a given time step (written at the end of a simulation)	$\text{Nb} \cdot \text{km}^{-2}$	default

Each file contains a header and the time series of the two-dimensional field (see Appendix C.3 for details on file format and structure). Hence, the dimensions of the variable written in the DYM file is Zlevel \times Nlat \times Nlong where Nlat, Nlong, Zlevel are the latitudinal and longitudinal dimensions and the number of time steps.

Population density by life stage

For classification of life stages in SEAPODYM see Table 1.1. The population density for larvae and juveniles is computed as the sum over all age classes within each life stage:

$$N(t, x, y) = \sum_a N(a, t, x, y)$$

with $N(a, t, x, y)$ the density of fish of age a in Nb/km² (model units of the state variables), at the coordinates (x, y) and at time t . Age a is either equal to 0 for larvae, or from 1 to the last age in juvenile life stage.

For young and adults the biomass density is multiplied by the mean weight, $w(a)$, of each age class and so the unit becomes mt/km². The mean weights can be found in the parfile (see Table 3.7). So, for the biomass density we have:

$$N(t, x, y) = \sum_a w(a) N(a, t, x, y)$$

with $a \in [\text{age_autonomous to age_maturity-1}]$ for young, and $a \in [\text{age_maturity to } A^+]$ for adults. Note, the numerical solution of model 1.1, that is spatial distributions $N(a, t, x, y)$, corresponds to the end of the time step (see Chapter 2, sections 2.1.2 and 2.2.2).

Movement rates

The velocity vector components (both passive and active) and the diffusion rate varying with habitat index are saved only for the last age class, A+. The speed of population density at age A+ is computed from the advection velocity field $\mathbf{v}(t, x, y) = (u(t, x, y), v(t, x, y))$ as:

$$V(x, y, t) = \sqrt{u(t, x, y)^2 + v(t, x, y)^2}$$

where $u \in \mathbb{R}$, $v \in \mathbb{R}$, $V \in \mathbb{R}^+$, and $D \in \mathbb{R}^+$.

Predicted and observed catches

The predicted catches are computed in the model and written in units defined in the parfile (see section 3.4.5), hence either in metric tonnes or in number of fish. Note, currently the model application does not support different units by fishery.

Habitat indices

Only two habitat files are saved: one at the age of first 50% maturity, and another for the oldest adults. In order to output habitats for all or selected age classes, use the `seapodym_habitats` application (see section [3.5.2](#)).

Restart

The date `yyyy-mm` corresponding to the last date of a simulation run is stored only in the file name. Since age dimension is used instead of time in the DYM file, the Zlevel vector contains age information. This restart file can then be used as the `init file`([3.4.4](#)). It is the user's responsibility to manage the time stamp of the state vector.

3.6.3 ASCII output files

Here is a list of the ASCII files generated by a simulation:

- `SumDym.txt` : for every time step of a simulation, total primary production, total biomass of micronekton groups, total biomass of population by life stage, fishing effort by fishery, observed and predicted catch by fishery.
- `SumQArea.txt`: for every time step of a simulation, total population biomass by life stage and by region.
- `SumEEZ_ID.txt`: for every time step of a simulation, total population biomass by life stage and by EEZ.
- `spname_MeanVar.txt`: spatial mean of mortality, advection and diffusion rates by age class.
- `spname_Spatial_Corr.txt`: spatial correlation between predicted and observed catch.
- `spname_LF_obs.txt`: observed length frequency by region aggregated over the whole time series in 4 quarters and in total.
- `spname_LF_Q_fishery.txt`: length frequency of catch by fishery and by quarter.
- `spname_LF_Q_sum.txt`: catch by age, by region and by fishery; quarterly sums for overall time period.

The following sections describe the contents of each file, while the file structure can be found in Appendix [C.2](#).

SumDym.txt

This file gives the time evolution of model variables aggregated over the entire domain. In other words, if we denote $\phi(x, y, t)$ the variable of interest, the file contains $\sum_{x,y} \phi(x, y, t)$.

Table 3.16: Configuration parameters and variables written in the **SumDym.txt** file.

Parameters	Description
<code>date</code>	The date of the time step given as <code>yyyy-m-d</code>)
<code>tstep</code>	The time step
Variables	Description
<code>P in C-C</code>	The total primary production, in $\text{mmol C}\cdot\text{m}^{-2}\cdot\text{day}^{-1}$, in the regions bounded by the latitudinal coordinates $C - C$. There are three regions: 10N–45N, 10S–10N and 35S–10S
<code>P total</code>	The total primary production over the area within $C - C$ coordinates, in $\text{mmol C}\cdot\text{m}^{-2}\cdot\text{day}^{-1}$.
<code>F_xxx</code>	Total biomass in mt of xxx-pelagic functional group of micronekton (among meso, hmeso, epi, bathy, mbath, hmbathy).
<code>B life_stage spname</code>	Total population biomass in mt by life stage (larvae, juvenile, recruitment, young, adult)
<code>B total spname</code>	Total population biomass in metric tonnes (mt).
<code>effort fisheryID</code>	Effort by fishery, in units provided for fishery with a short name <code>fisheryID</code> .
<code>obs C_spname _fisheryID</code>	Total observed catch by fishery in mt or Nb
<code>pred C_spname _fisheryID</code>	Total predicted catch by fishery in mt or Nb
<code>obs CPUE_spname _fisheryID</code>	The observed Catch Per Unit Effort by fishery, in units depending on units of catch and effort
<code>pred CPUE_spname _fisheryID</code>	The predicted Catch Per Unit Effort by fishery, in units depending on units of catch and effort

SumQArea.txt

This file stores the predicted population abundance (either as total number of fish or as total biomass) aggregated over the rectangular regions defined in the parfile (see section 3.4.7). The population abundance is calculated for different life stages (for classification of life stages in SEAPODYM see Table 1.1). Providing the age at recruitment to the exploited stock and the regional structure as defined in a given stock assessment model, these aggregated outputs can be used for comparisons with the stock assessment model outputs. The variables in the

file are detailed in Table 3.17. See the structure of this file in Appendix C.2.2.

Table 3.17: Configuration parameters and model variables written in the `SumQArea.txt` file

Parameters	Description
<code>regional coordinates</code>	The lon-lat coordinates of the corners of rectangular regions defined in the <code>area</code> node in the parfile.
<code>life stage</code>	The indices of the age classes in each life stage
<code>date</code>	Date written as <code>year</code> , <code>month</code> and <code>day</code> columns
Variables	Description
<code>spname N life_stage region r</code>	Total number of fish by life stage (from larval stage to recruits) in region r
<code>Total N life stage</code>	Total number of fish from larval stage to recruits over all regions
<code>spname B life_stage region r</code>	Total biomass of fish by life stage (for immature and mature adults) in region r
<code>spname B tot. region r</code>	Total biomass of fish at adult stage (immature and mature combined) in region r .
<code>Total B</code>	Total biomass of adult fish over all regions

`SumEEZ.txt`

These files have exactly the same variables as `SumQArea.txt`, but the EEZ area is used for biomass aggregation instead of rectangular regions.

`MeanVar.txt`

This file, used for diagnostics, contains the spatial mean of different two-dimensional fields computed but not written in DYM files. The contents of the file are listed in Table 3.18. Each variable is provided by age class, from 0 to the last A^+ age class.

Table 3.18: Configuration parameters and variables written in the `spname_MeanVar.txt` file.

Parameters	Description
<code>date</code>	Written in <code>year</code> , <code>month</code> , <code>day</code> columns
Variables	Description
<code>mortality-at-age p</code>	Mean mortality rate (mo^{-1}) of the species <code>spname</code> at age a_p .
<code>speed p</code>	Mean speed ($\text{nmi} \cdot \text{mo}^{-1}$) of the species <code>spname</code> at age a_p .
<code>diffusion p</code>	Mean diffusion rate ($\text{nmi}^2 \cdot \text{mo}^{-1}$) of the species <code>spname</code> at age a_p .
<code>temperature p</code>	Mean water temperature ($^{\circ}\text{C}$) weighted by the population density at age a_p .

Spatial_Corr.txt

The spatial correlations between the observed and predicted catches as well as Student test probability values are computed at each time step for all fisheries and written in file `Spatial_Corr.txt` (Table 3.19).

Table 3.19: Configuration parameters and variables written in the `Spatial_Corr.txt` file.

Variables	Description
<code>n</code>	Number of values used to compute correlation.
<code>r_fishery spname</code>	Correlation between observed and predicted catches by fishery.
<code>prob</code>	Student estimation.
<code>CPUE_r_fishery spname</code>	Correlation between the observed and the predicted CPUE by fishery.
<code>prob</code>	Student estimation.

spname_LF_obs.txt

This file gives both the observed catch-at-age and the sum of catch-at-age per fishery, region and quarter (Table 3.20).

Table 3.20: Configuration parameters and variables written in the `spname_LF_obs.txt` file.

Parameters	Description
Length	Mean lengths in age classes, in cm.
Variables	Description
<code>f_spname_region_r</code>	Observed catch-at-age by fishery f and region r .
<code>sum_f_spname</code>	Total observed catch-at-age by fishery f over all regions.
<code>sum_spname_region_r</code>	Total observed catch-at-age for all fisheries in region r .

`spname_LF_Q_fishery.txt`

This file contains the predicted length frequencies aggregated over model age classes (catch-at-age) per quarter, per fishery and per region (Table 3.21). It has the same tabular format as the input LF data file (see Appendix sections C.1.2 and C.2.7).

Table 3.21: Configuration parameters and variables written in the `spname_LF_Q_fishery.txt` file

Parameters	Description
Fisheries	Fishery ID
Regions	Region ID
date	Date written in the columns Year, Quarter and Month
Variables	Description
<code>LF[p]</code>	Predicted catch-at-age in number of fish of age a_p caught by fishery in a given region

`spname_LF_Q_sum.txt`

Together with the file `spname_LF_Q_fishery.txt`, this file is mostly used for model validation, as it provides detailed information on the age distribution of catches by fishery, region and quarterly time intervals predicted by the model. Note, quarter is the time interval at which the LF data are collected.

The row with the total catch-at-age data corresponding to the mean length l , for quarter q of the column `fisheryID_spname_region_r` is computed from the sum of the values written

in the file `spname_LF_Q_fishery.txt` for corresponding date D , region r , fishery f and quarter q :

$$\sum_{D \in [q_i \cap r \cap F]} \text{LF}[l]_D.$$

Table 3.22: Configuration parameters and variables written in the `spname_LF_Q_sum.txt` file.

Parameters	Description
<code>length</code>	The mean length in cm of the age class.
Variables	Description
<code>fisheryID</code>	Predicted catch-at-age by fishery in number of individuals (Nb)
<code>_spname_region_r</code>	of age class with mean length l , in region r during quarter q
<code>sum_fisheryID</code>	Total predicted catch-at-age (Nb) by fishery over all regions during
<code>_spname</code>	the quarter q
<code>sum_spname</code>	Total predicted catch-at-age (Nb) by all fisheries in region r during
<code>_region_r</code>	quarter q

References

- Lehodey P., André J-M., Bertignac M., Hampton J., Stoens A. 1998. Predicting skipjack tuna forage distributions in the Equatorial Pacific using a coupled dynamical bio-geochemical model. *Fisheries Oceanography* 7: 317–325.
- Lehodey, P., Murtugudde R., Senina I. 2010. Bridging the gap from ocean models to population dynamics of large marine predators: A model of mid-trophic functional groups. *Progress in Oceanography*, Volume 84:69-84.
- Lehodey P., Conchon A., Senina I., Domokos R., Calmettes B. et al. 2015. Optimization of a micronekton model with acoustic data. ICES Journal of Marine Research. 53, 571-607. Science 72(5):1399–1412. <https://doi.org/10.1093/icesjms/fsu233>

Chapter 4

Model parametrisation

Before applying the model to solve fishery management problems we need to be confident of the reliability of model predictions; hence, we first need to work on achieving the model dynamics under tight coupling of model predictions to observations. In the fields of numerical modelling of physical phenomena in atmosphere and ocean, this is referred to as ‘data assimilation’, and in ecology it is called quantitative modelling; the purpose of both is the same – to provide the estimates of nature, which are better estimates than can be obtained by using only the observational data or only the dynamical model (Robinson and Lermusiaux, 2002). Hence, quantitative modelling relies on integrating as much available data as is necessary to observe (and constrain) model dynamic processes through all considered dimensions. In fish population dynamics modelling, we do not directly observe the model variables, that is, fish biomass, so we cannot use inverse methods to correct the model state as is done in meteorological or ocean circulation models. Instead, we use inverse methods to compute the gradient of the function of model parameters and observations with the aim to inform model parameters from the data, which contains observations for different processes of fish population dynamics. As an outcome, we want to achieve an optimal model parametrisation providing the model (i) describes the assimilated data and (ii) remains valid for independent data sets. The latter can be fulfilled through space–time invariance of model parameters. Indeed, the SEAPODYM parameters, proper to the modelled species and its physiological characteristics, do not include dependencies on its environment. In contrast, they describe the species response to the environment that is included in the model as the forcing data. This chapter describes methods implemented in SEAPODYM to assist in the development of quantitative models. These methods, however, should not be considered as ready-to-use or universal recipes for building quantitative models for the new species with SEAPODYM as this process involves solving highly dimensional non-linear optimization problems and the success depends strongly on the amount of observations available.

4.1 Integrating data in a fish population dynamics model

Since SEAPODYM is a spatio-temporal model with age dynamics, describing the entire population life cycle, it possesses high dimensionality that inevitably requires the use of massive geo-referenced datasets. For tuna species we use three types of geo-referenced data: (1) fisheries data, including effort and catch aggregated in time and spatial area; (2) length frequency sample data provided at usually coarser resolutions, such as season and large regions; and (3) conventional tagging data including the positions of release and recapture as well as size characteristics of individual fish. It is important to note here that fisheries data must account for as close to 100% of fishing mortality as possible (Figure 4.1) to enable unbiased estimation of population abundance, but it is not required that all fisheries data are used in the parameter estimation method.

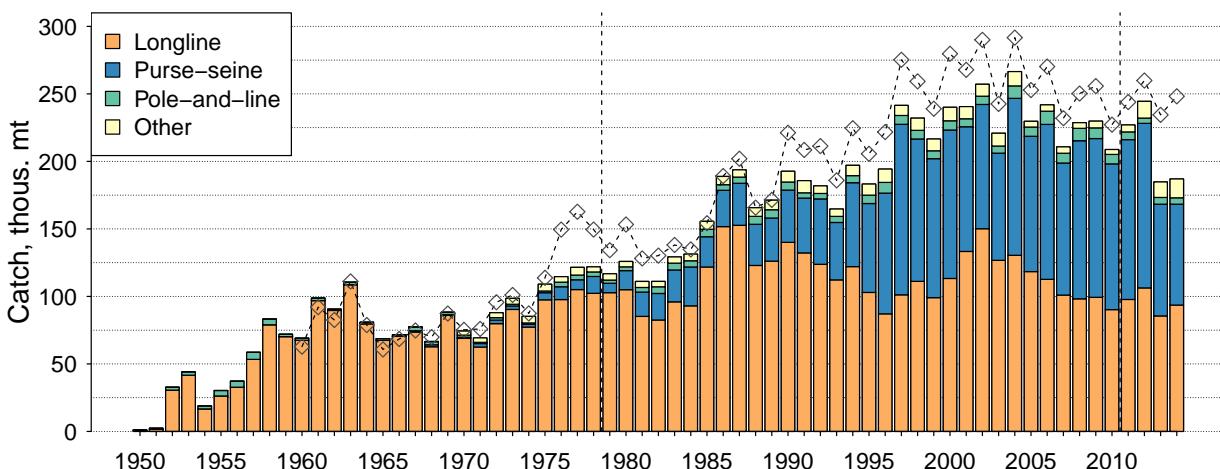


Figure 4.1: Total annual bigeye tuna catch aggregated from geo-referenced catch (Pacific-wide) used in SEAPODYM analyses. The dashed line corresponds to total landings of bigeye tuna (SPC Year Book, 2016).

4.1.1 Effort and catch data

Geo-referenced fishing data include effort and catch by flag, gear and fishing strategy. These data on tuna and billfish are collected by Tuna Commissions and are publicly available to a certain level of detail. The following Tuna Commissions contributed to the SEAPODYM studies: the Inter-American Tropical Tuna Commission (IATTC) for the eastern Pacific Ocean, the Pacific Community (SPC) on behalf of the Western Central Pacific Fisheries Commission (WCPFC) for the western and central Pacific Ocean, the Indian Ocean Tuna Commission (IOTC) for the Indian Ocean and International Commission for the Conservation of Atlantic Tunas (ICCAT) for Atlantic Ocean. The large industrial fishing fleets

targeting different tuna species use several gears – purse-seine (PS), pole-and-line (PL), longline (LL) and troll (T). Typically, for tuna species, monthly spatially distributed data on large-scale industrial and small-scale domestic fishing are available for effort, $E_{t,f,lon,lat}$, as a number of sets, days fishing or a number of hooks; and catch, $C_{t,f,lon,lat}^{obs}$, in weight or number of fish (see Fig. 4.2), where indices t, f, lon, lat refer to the year-month, fleet, and geographic coordinates of 1-degree cells.

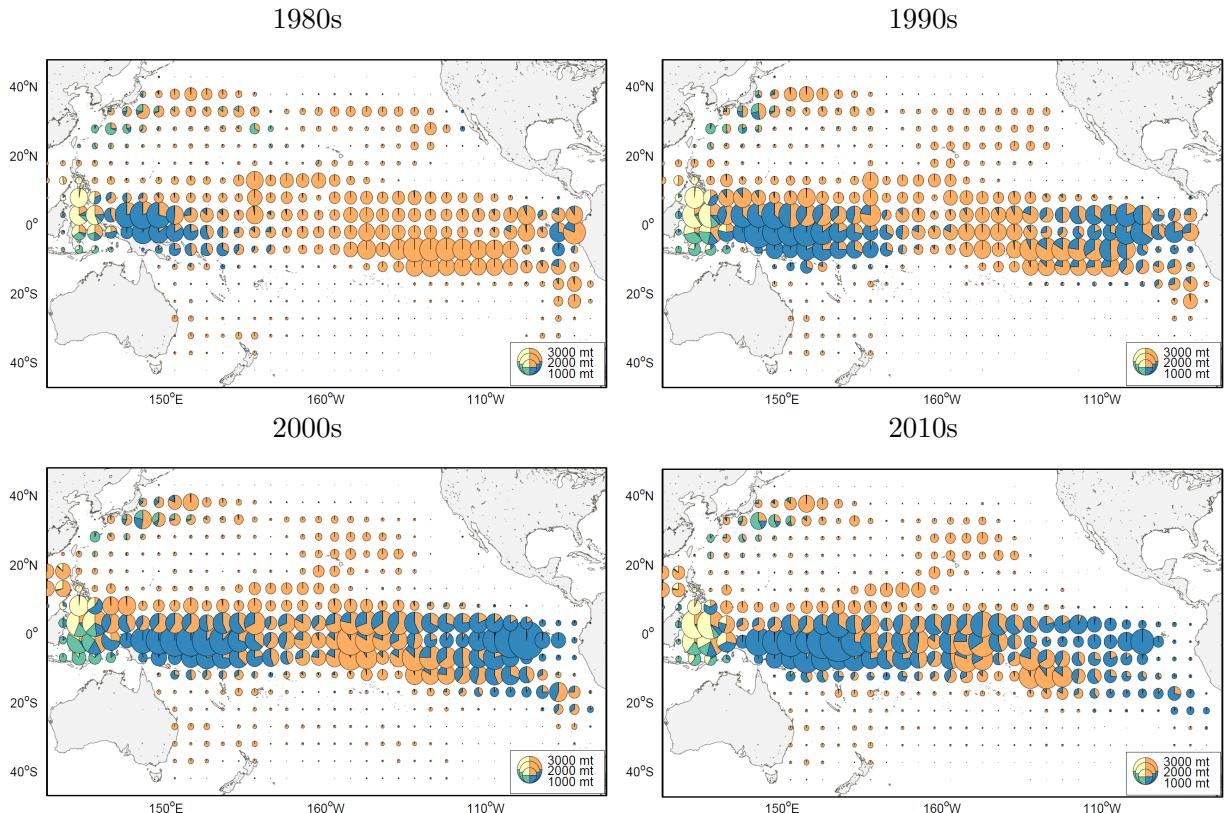


Figure 4.2: Spatial distributions of bigeye tuna catches by decade and by gear: longline (orange), purse-seine (blue), pole-and-line (green), and others (yellow).

Prior to integrating these data into SEAPODYM for parameter estimation, it is important to organise them into homogeneous fisheries. The latter are defined in SEAPODYM by a unique selectivity function, which does not depend on space and time, and by a unique catchability coefficient, which can only change linearly with time. The linear trend may be necessary to account for technological advances, affecting the time of searching and volumes of fish being caught. Spatial and temporal variability in catch is assumed to be driven by the spatial distributions of fish that are explicitly described by the model. Another source of variable catchability, oceanographic features such as fronts and eddies, are accounted for as well through the habitat and movement modelling. The habitat index reflects both thermal

gradients and the concentrations of prey species, transported and organised into heterogeneous spatial structures by ocean currents. Hence, the gradual changes in fisheries, such as geographic expansion or seasonal displacements of fishing grounds, and the continuous increase in fishing effort and catch (Figure 4.2) can be successfully integrated. In contrast, all abrupt changes in fisheries parameters due to changes in fishing strategy and target species, introduction of regulatory measures, whether seasonal or not, need to be eliminated by splitting the fleet data. Usually, this information can be derived either from Commissions' reports, from the provided fisheries descriptions, or by conducting the time series analysis. Finally, the data with unreliable, partially available fishing effort should also be removed from fisheries whose data will be integrated within the parameter estimation method. Such fisheries can be accounted for with the help of the catch removal method described below.

Catch prediction methods

Previous studies with SEAPODYM using fisheries data to inform model parameters indicated several problems with the use of fishing effort, in particular those for purse-seine gear. Using fishing effort to predict the catch in each observed spatial position logically results in fish distributions with the centre of mass shifted to the areas with highest CPUE and the lowest biomass in areas with the lowest CPUE. The highest CPUE can be due to highest biomass, but also to lowest effort and in purse-seine data, where the fishing effort, expressed in days, may include time unrelated to fishing, it is often observed that the CPUE is lowest in the main fishing grounds, and highest on the edges of the fishing grounds. Fitting the model predictions by relying on such effort leads to biases and excessive dispersal and hence overestimation of fish density away from the observed locations, creating a large quantity of non-observed ‘cryptic’ biomass, which might be unrealistic. The term ‘cryptic’ biomass was first introduced by Fonteneau (1996) for the fraction of the stock that is unavailable to fisheries. To avoid such biases, there are two catch prediction methods in SEAPODYM: the classical Gordon-Schaefer formula and the catch removal method (see also the fishing mortality section 1.2.7).

The first method is based on the fishing effort. The total mortality of all fish of age a due to fishing, $m_F(a)$, eq. 1.42, depends on the catchability coefficients, q_f , of fisheries f , their selectivity, s_{fa} , for the fish at age a , and the observed fishing effort, E_f . At every spatial location and time, the instantaneous total catch C in number of fish of age a is computed as follows:

$$C(a) = m_F(a) \cdot N(a) \cdot A_{xy}, \quad (4.1)$$

where $N(a)$ is modelled density of fish in number of individuals per unit area (Nb/km^2), and A_{xy} is the area in km^2 . However, observations are available for the large time intervals and spatial area. Passing to numerical model notations (see sections 2.1.1–2.1.3), we define

C_{fKIJ}^{pred} , the total predicted catch in mass units (mt – metric tonnes) by fishery f at outer time step K and observational grid indices (I, J) as:

$$C_{fKIJ}^{\text{pred}} = q_f E_{fKIJ} \sum_{p=0}^{n_a} s_{fp} w_p \sum_{ij \in IJ} N_{pKij} \Delta x_i \Delta y_j, \quad (4.2)$$

where w_p is the mean weight of fish and N_{pKij} is the density of fish in the p -th age class and numerical grid indices (i, j) . As seen in this formula, the effort data should be provided at the model's temporal resolution; however, the spatial resolution of observations can be retained. It allows the effort to be applied to the model biomass after its aggregation to the larger area, and the predicted catch to be compared to observed catch at its original resolution. Note that the prediction of catch rate, that is, catch per unit of effort by fishery, is straightforward with this catch prediction method. The prediction of catch per unit of effort at the resolution of observations, $\text{CPUE}_{fKIJ}^{\text{pred}}$, is obtained by division of eq. 4.2 by the effort value, E_{fKIJ} .

Thus, the predicted catch depends not only on the available fish biomass, but also on the fishing effort, gear catchability and selectivity. Since the selectivity parameters do not vary in space and time (with catchability either constant or allowed to change linearly over time), the estimation of population spatial distribution in the MLE approach is essentially driven by the spatio-temporal variations of the biomass and the fishing effort. Unfortunately, this approach is correct only in the ideal case, that is, when the fishing effort is well estimated and fully reported. For gears such as purse-seine, the estimation of fishing effort is problematic as one should take into account not only the number of sets that were employed after the fish school had been detected, but also the time the boat spent in active searching. The second variable is often hard to estimate and usually the effort becomes inflated in the principal fishing grounds characterised as zones with the highest catches and where the fishing boats spend more time and is underestimated in the areas, which are rarely visited by the fishers. As an example, the observed CPUE of purse-seine fisheries targeting free schools of skipjack tuna are higher outside the main fishing grounds, thus showing a clear density gradient towards their edges. One can also imagine that using equation 4.1 in MLE in the case when the catch is highly correlated with the effort will lead to estimating a homogeneous distribution of biomass. Another problem can arise with the effort of those fisheries that do not target the modelled species or change the target during their fishing campaign. If the information about currently targeted species is not contained in the fishing effort, the biomass estimation becomes biased because zero catch does not necessarily reflect the absence of the non-target species.

To avoid the biases associated with the use of inaccurate fishing effort, another method was implemented in SEAPODYM. It consists of removal of total (summarised over all fisheries) catch in number of fish of a given age class directly from the predicted fish density. Let us denote N_{pkij} as the population density in number of fish of age class p per unit area at

the model integration step $k \frac{\Delta T}{n}$ ($k = 1 \dots n$, n – number of iterations in the ADI method); $C_{pkij}^{\text{obs}} = \frac{1}{n} C_{pKij}^{\text{obs}}$ is the n th portion of the total observed catch at time interval (T_K, T_{K+1}) . Then the corresponding predicted catch at the ADI method sub-step is computed as follows:

$$C_{pkij}^{\text{pred}} = \min \left(C_{pkij}^{\text{obs}}, (N_{pkij} \cdot \Delta x_i \Delta y) \right). \quad (4.3)$$

Note, that according to eq. 4.3 the predicted catch is either equal to or less than the observed one. The latter occurs only if there is not enough biomass in the model grid cell. Finally, the predicted total catch is summed up through n ADI iterations and the total value is attributed to the fisheries f operating in the cell ij according to their observed ratio in this cell, that is,

$$C_{fpKij}^{\text{pred}} = \frac{C_{fpKij}^{\text{obs}}}{\sum_f C_{fpKij}^{\text{obs}}} \sum_{k=1}^n C_{pkij}^{\text{pred}},$$

gives the catch by fishery at model temporal and spatial resolutions. So, in contrast to the first method, this approach requires data redistribution, which is done automatically in the model software.

Equations 4.3 and 1.44 can be combined with the classical eqs. 4.2 and 1.42 by applying the Gordon-Schaefer formula to selected fisheries, for which the fishing effort can be considered unbiased (longline and pole-and-line fisheries targeting modelled species). One inconvenience of the catch removal approach is that it cannot combine catch data with different units (e.g., metric tonnes and numbers of fish).

4.1.2 Length frequency data

The length frequency (LF) data are the geo-referenced sample data, including information on flag, gear, fishing method, temporal interval, region and sampled catch at length (raised or not), and the total catch by a given fleet. However, these data are usually aggregated by quarter and over larger areas, for example, $5^\circ \times 5^\circ$, $5^\circ \times 10^\circ$, $10^\circ \times 10^\circ$ or larger (see Figure 4.3). Note, the fisheries structure in LF data must be conformal to that in the effort and catch dataset.

Length frequency prediction

By definition, the length frequency is the proportion of fish of size $\ell(a)$ observed in the catch by fishery f at time t and in region r . Since SEAPODYM is an age-structured model, we need to redistribute the catch-at-length to catch at the mean length of model age classes, which is in most cases straightforward as the model age classes usually have a coarser resolution than

the observed length bins. In discrete space and time notation, the predicted LF, denoted Q^{pred} , is computed as follows:

$$Q_{fpKr}^{\text{pred}} = \frac{s_{fp} \sum_{i,j \in r} E_{fij} N_{pKij} \Delta x_i \Delta y_j}{\sum_{p=0}^{n_a} s_{fp} \sum_{i,j \in r} E_{fij} N_{pKij} \Delta x_i \Delta y_j} \quad (4.4)$$

where selectivities s_{fp} are described in eq. (1.43), $ij \in r$ are the indices of the model grid, region r is defined by observations, Δx_i and Δy_j are grid cell sizes as defined in section 2.1.1.

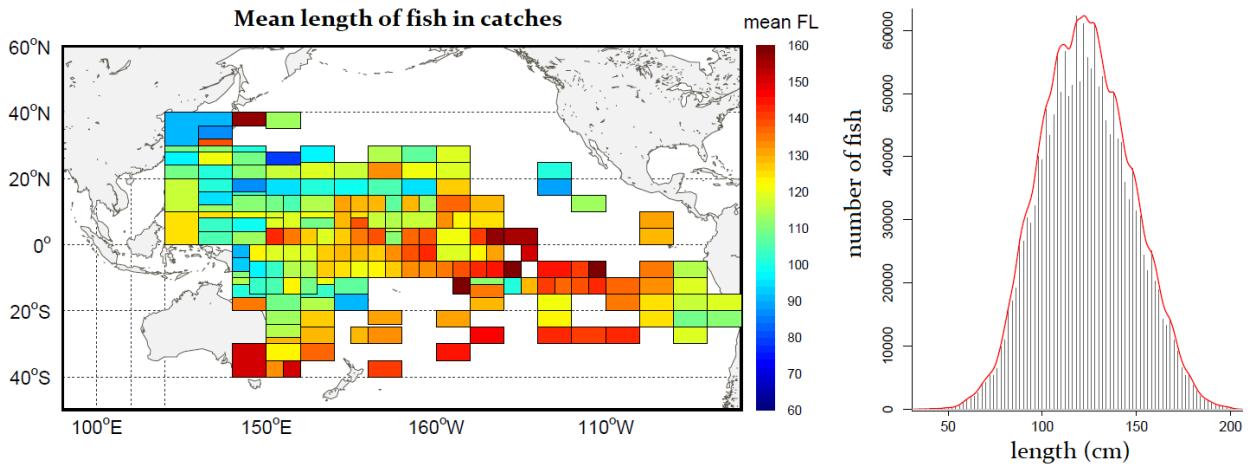


Figure 4.3: Typical length frequency dataset comprising distributions of fish sizes in catch withdrawn in multiple sample regions. The colour on the left-hand side of the map indicates the mean fork length (FL) of fish caught in each region. The vertical lines on the histogram show the sampled number of fish at a given size. The solid red line corresponds to the kernel density estimation for the data sample.

4.1.3 Conventional tagging data

Conventional release-recapture tagging data include information on position, date and size of the fish upon release and the same information upon fish recapture (Figures 4.4 and 4.5). Tagging campaigns are regularly conducted in the Pacific Ocean by SPC and IATTC under different tuna tagging programs, which started as early as 1967. In particular, SPC has conducted several large tagging experiments since the 1980s in the western and central Pacific Ocean (WCPO). For example, since 2008 SPC scientists deployed about 200,000 conventional tags on skipjack tuna. IATTC and the Japanese Fisheries Agency have been also very active in tuna tagging in the eastern and north-west Pacific respectively. While releases are made by fisheries scientists, the recaptures are by fishermen. Usually hundreds

of tunas are released in a single location, and the spread of recapture locations depend on the fishing activity. Obviously, only a small proportion of the released fish is recaptured. The major difference between release and recapture data is its reliability. The data on releases are fully reported, while recordings of recapture locations and dates as well as the fish size upon recapture are subject to error. Therefore, special care is needed in preprocessing the tagging data, that is, verifying the boat positions through Vessel Monitoring System (VMS) data, and correcting or filling a gap on size information using known growth models for the species.

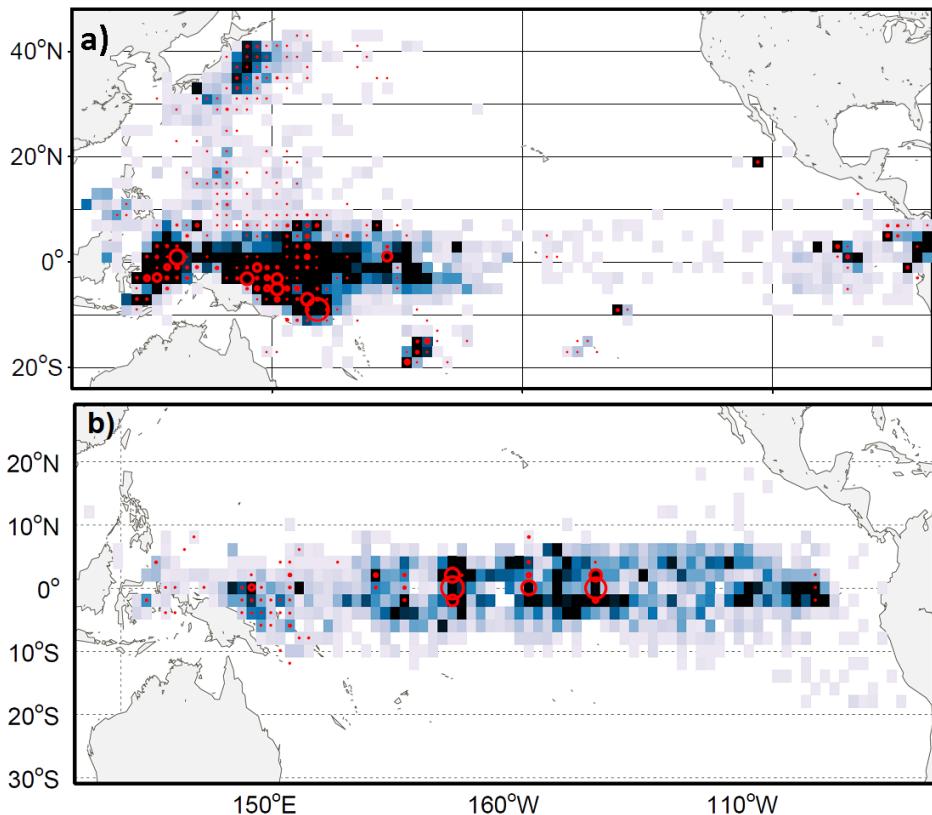


Figure 4.4: Number of skipjack (a) and bigeye (b) tunas that were tagged during conventional tagging campaigns (size of red circles is proportional to the number of released fish) and recaptured by industrial fishing vessels. The 1° pixels are coloured with a linear colour scale from white to blue indicating 0 to 40 and more tag returns respectively.

There are other caveats in the use of conventional data. First, they do not allow observing of the entire population, but essentially the juveniles and immature adult fish that are associated with surface schools and can be caught with surface gears. For example, only juvenile and immature bigeye tunas were tagged and released along the equator (bottom panels in Figures 4.4 and 4.5). Second, a majority of all tagged tunas are recaptured shortly

after release. For bigeye tunas, 89% of all recaptured fish were recaptured within one year of liberty at sea, 10% between one and two years and only 1% after more than two years. Consequently, with size at 50% maturity being 115 cm, the majority (93%) of recaptures are still immature tunas. Hence, observing only part of the population may present difficulties for estimating model parameters responsible for the dynamics of an unobserved fraction of the population. Third, unknown reporting rates and strong dependence of tag recaptures on the associated geo-referenced fishing effort make it difficult to rely on these data in the estimation of natural mortality rates.

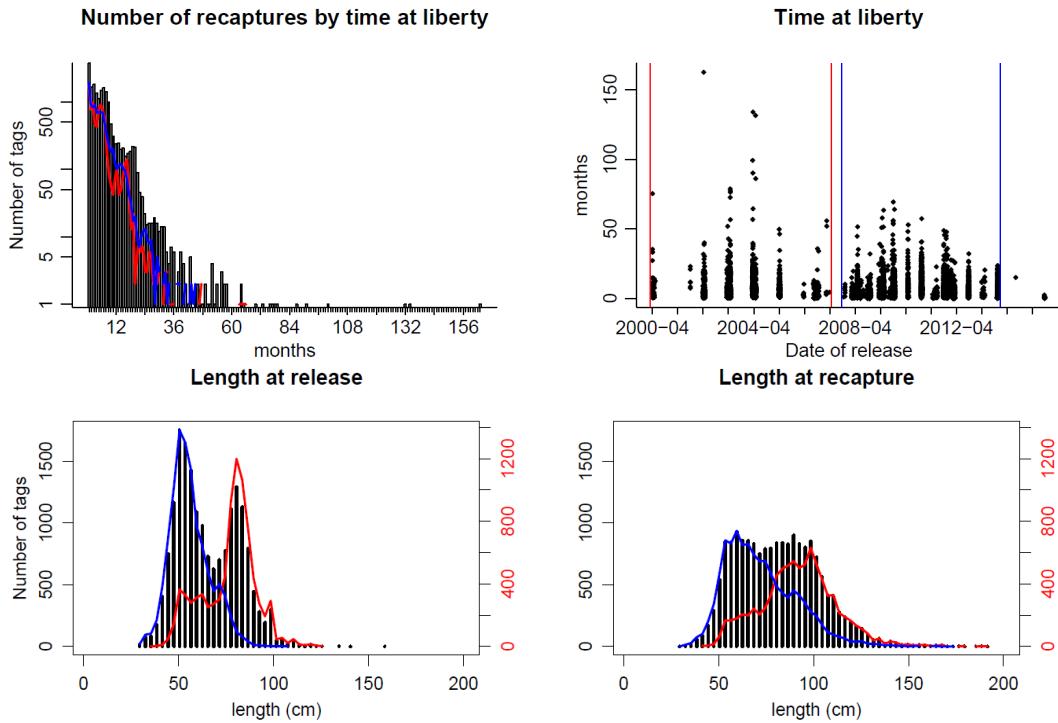


Figure 4.5: Available conventional tagging showing two distinct periods (enclosed by red and blue lines) in tagging data with different statistical properties of fish distribution. The histogram and scatter plot on time at liberty show the time at liberty frequency diagram and the time at liberty of the tags depending on their date of release, respectively. The size distributions at release and recapture depict the size (age) intervals that are observed by tagging data.

Conventional tagging data (Figure 4.4) are integrated into the optimization method in SEAPODYM essentially to improve the estimates of habitat and movement parameters that are critical to control the overall population dynamics. Therefore, the approach considers only fish that have been recaptured (see next section for more details), as these are the only data that contain information about potential movement (Senina et al., 2020b). Nevertheless,

compiled datasets of releases and recaptures, provided by SPC and IATTC from different tuna tagging programs, contain tens of thousands of records on released and consequently recaptured fish.

For the purposes of reducing computational costs and integrating maximum information to inform dynamic processes through all model dimensions, it is important to choose the time period for data to be used by data assimilation methods. The criteria to be followed in making this choice are (1) the subset covers the time period without large gaps, (2) the subset does not reduce the observed time at liberty, and (3) the lifespan coverage of tagged fish is the same as for the whole dataset. For example, the tagging data temporal coverage and the distribution in terms of mean length and time at liberty illustrated in Figure 4.5 reveal two characteristic periods with massive tagging of bigeye tuna within the 2000–2013 time range, clearly distinguished by the length of released tunas, the positions of release and the distributions of recaptures. During the first one, from early 2000 to mid-2007, the mean length of bigeye tuna at release was 77 cm; they were mostly tagged at three release positions around the equator at 95°W longitude and a few dozen tunas were tagged and recaptured in the warm pool area. During the second period, from mid-2007 to the end of 2013, much smaller bigeye tuna, 56 cm on average, were tagged, extending the area of release towards the central Pacific Ocean. So, providing a better coverage for spatial and age structure, the second subset of recaptures (2008–2013) constituting 50% of the dataset (shown in Figure 4.4), panel (b) and in blue in Figure 4.5) is a better candidate for informing model parameters, while the first can be left for validation.

Movement model for tag prediction

The tagged fish represent a subset of the modelled population, and hence we can apply the same mathematical equations to describe their spatio-temporal dynamics. The approaches suggested earlier for Eulerian models (Sibert et al., 1999) or used in the statistical stock assessment models (see e.g., Hampton and Fournier, 2001) rely on release–recapture data to inform estimations of both movement and natural mortality parameters. However, this approach is not practical within SEAPODYM. First, it leads to the augmentation of the model state vector by a large number of variables to describe the spatial dynamics of a tagged sub-population within the Eulerian model as we need to separate it into cohorts of fish being tagged and released at the same time and in the same area (Sibert et al., 1999). Second, tag reporting rates (the proportion of recaptured tags that are reported by fishermen) are not known. It means that a large proportion of tags might be recaptured and not reported, hence providing biased information on natural mortality. Calibrating the reporting rates within the parameter estimation is difficult because of their variation in space and time. Third, prediction of tag recaptures in this model depends on the quality of geo-referenced fishing effort and gear information. If these difficulties are not properly addressed, the information provided by tags on natural mortality may not ensure unbiased parameter estimation. At the same time, the main objective of integrating tagging data within SEAPODYM is to

improve model predictions by adding information on fish movements. Hence, we must focus on modelling the movement of tagged fish. Since only recaptured tags can be informative of movement, we use only those released fish that were recaptured and reported. Because fish movements are essentially driven by the habitat indices based on environmental variables, the tagging data provide key information to estimate both habitat-related and movement parameters.

Tags recaptured in the same month (quarter) are aggregated into cohorts. In other words, instead of defining the cohort by the same origin, it is defined by the same ending, that is, tags are aggregated into cohorts by their common time of recapture. As one cannot trace the density in an ADR model, this allows precise accounting for the time at liberty of all tags. Integration of tagging data within the population dynamic model (1.1–1.3) consists of adding the model of tag movement that is driven by the same model mechanisms and parameters. Let $R_k(a, t, \mathbf{x})$, $a = a_n, \dots, a_{n+\tau}$ be the density of the k th cohort of tagged fish of age a , varying between the age at release a_n and age at recapture $a_{n+\tau}$, where time at liberty, τ , is known from the recapture data. The variables R_k are defined in two-dimensional space $\mathbf{x} \in \Omega \subset \mathbf{R}^2$ and the time $t \in (t_0, t_R)$, where t_0 is the time of release of the first tag in the cohort k and t_R is the time of recapture of all tags in the cohort. The model of movement of the k th cohort of tag density is based on a system of advection–diffusion equations with ageing and source terms, zero initial and zero-flux boundary conditions:

$$\begin{aligned}\partial_t R_k + \partial_a R_k &= -\text{div}(\mathbf{v} \cdot R_k) + \nabla(D\nabla R_k) + r_k & (4.5) \\ R_k(a, t_0, x) &= 0 \\ \mathbf{n} \cdot \mathbf{v} \Big|_{\mathbf{x} \in \partial\Omega} &= \mathbf{n} \cdot \nabla R_k \Big|_{\mathbf{x} \in \partial\Omega} = 0\end{aligned}$$

where advection \mathbf{v} and diffusion D rates are the same as in the model 1.1, and the source term $r_k(a, t, \mathbf{x})$ includes all releases of age a into cohort k at time t .

The model 4.5 with parameters of the model 1.1, included in the MLE approach, allows the estimation of movement and habitat parameters of model (1.1–1.3). Obviously, such an approach does not allow the estimation of natural mortality from tagging data, but presents several advantages. Due to the precise time at liberty in 4.5, the proposed method allows significant reduction of the size of the model state vector. When it is not known, for example, in the classical tag attrition models, the use of tagging data within the Eulerian population dynamics model implies considering a large number of cohorts of fish aggregated by their age, position and time of release. The latter leads to an increase of model dimension and hence the time and memory required by the PDE numerical solver and adjoint method, making the classical approach too costly. Finally, since the probability of recapturing all tags is 1 in the model 4.5, this method omits the use of fishing effort, fisheries parameters and reporting rates to predict tag recaptures by fishers. Hence, it is independent of fishing data and particularly well adapted to computationally demanding parameter estimation in the SEAPODYM model.

Nevertheless, using only reported and recaptured tags, the computational cost of adding tags in the MLE may increase sharply with the number of tagged cohorts, that is, the number of equations 4.5. Hence selecting a subset with the same properties as the whole dataset in terms of spatial, temporal and age coverage, might be practical to reduce the time interval (t_0, t_R) . The movement modelling in SEAPODYM is driven by the environment and the behavioural response to the environment, expressed via preferred habitat and movement parameters, is assumed to be constant over all model dimensions. That is why the estimation of habitat and movement parameters can be accomplished using the data subset, both in terms of space and time coverage.

Several difficulties need to be addressed while integrating individual movement data into the Eulerian model. Although many recapture positions can be checked with VMS data, the positions of observed recaptures cannot be considered as exact. To account for the uncertainty of tag recapture distributions, the number of observed recaptures distributed in two-dimensional space are interpolated using Gaussian kernels. This is necessary not only to account for uncertainty in reported positions, but also to relax constraints imposed on density distributions by individual movement. Also, before including in the likelihood, observed and predicted tags were aggregated into 6° cells over a 3-month period.

A bivariate Gaussian kernel for two independent variables (longitudinal and latitudinal coordinates) is applied to the observed recapture records to account for their uncertainty and to obtain smooth density fields of the recaptures that can be compared to the continuous fields of modelled densities in the likelihood framework:

$$K_G(x_R, y_R; \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{(x-x_R)^2}{2\sigma_x^2} - \frac{(y-y_R)^2}{2\sigma_y^2}} \quad (4.6)$$

where x_R and y_R are the coordinates of tag recapture, the values of σ_x and σ_y are related to the longitudinal d_x and latitudinal d_y displacements respectively as follows: $\sigma_x = \sqrt{\frac{d_x^2 \Delta t}{TL}}$ and $\sigma_y = \sqrt{\frac{d_y^2 \Delta t}{TL}}$ where Δt is the model time step and TL is the tag's time at liberty, assuming that the individual didn't move straight between release and recapture positions (represented by Euclidean distance) but did rectilinear displacements (Manhattan distance). To avoid values of σ_x and σ_y being too small or too big, they are restricted in the interval bounded below by 100 nmi and the maximal values are computed according to the maximal theoretical diffusion rate, linked to the size of individual. Finally, each kernel is rescaled to account for the lost density on land, and monthly distributions of the observed tag recaptures are represented by the sums of all kernel functions as follows:

$$R_{at}^{obs} = \sum_{k=1}^{n_t} K_G(x_{Rak}, y_{Rak}; \sigma_{xak}, \sigma_{yak}) \quad (4.7)$$

where n is the number of all tags recaptured in month t . Note that the age at release a_r is computed for each tag as the inverse of the Von-Bertalanffy growth function $a_r = VB^{-1}(L^{obs})$ using the measured fork-length, and the age of recapture is then $a = a_r + TL$.

One of the peculiarities of the tagging datasets is that the majority of the tags are recaptured by fishermen in the proximity of the release positions within the first months. To avoid informing the model with zero movements it is important to weight the observations in that likelihood in order to emphasise those that stayed longer at liberty. This can be done by introducing weights proportional to the time at liberty, computed simply as integrated time at liberty over all tags in the grid cell:

$$W_{tij} = \sum_{k=1}^n \left(R^{obs}{}_{kij} \cdot TL_k \right) / \sum_{k=1}^n R^{obs}{}_{kij}.$$

Also, to account for scarcity of tag recaptures, both observations and predictions are aggregated to bigger grid cells IJ and over a longer time period q before being included in the likelihood. We use $q = 3$ months and $5^\circ \times 5^\circ$ grid cells.

$$R_{qIJ}^{pred} = \sum_{t \in q, ij \in IJ} (W_{tij} R_{tij}^{pred}) ; R_{qIJ}^{obs} = \sum_{t \in q, ij \in IJ} (W_{tij} R_{tij}^{obs})$$

Note that the described aggregations of tags into bigger time and space strata and the smoothing of the distributions of tag recaptures using Gaussian kernels are both done automatically by the model software. The sizes of bigger I, J cells as well as the use of Gaussian kernels can be configured through the parfile (see Appendix A.2).

4.2 Sensitivity of model parameters

Sensitivity analyses are useful tools to reveal which parameters can be estimated from available data and which cannot. Therefore, it is important to perform sensitivity analysis before proceeding to parameter estimation. If model predictions are insensitive to some parameters, it is unlikely that they will be determined uniquely from available observations and they should, therefore, be removed from the optimization. Two types of sensitivity analyses can be performed in SEAPODYM – local and global sensitivity analyses.

4.2.1 Local sensitivity analysis

Sensitivity of model predictions only. The first type of analysis examines how the predictions of the model are sensitive to its parameters. For this purpose we simply need to construct a function of the model solution, which represents model predictions (Worley, 1991). Then, the measures of sensitivity can be computed using precise gradients obtained from adjoint calculations. Since two types of data are assimilated within the model, that is, catch and length-frequencies, we construct the following functions:

$$R_1 = \sum_{fKij} (C_{fKij}^{pred})^2, \quad R_2 = \sum_{fpKr} (Q_{fpKr}^{pred})^2, \quad r = 1, \dots, n_f. \quad (4.8)$$

Then we define two measures of relative sensitivity $\xi_1(\theta_k^0)$ and $\xi_2(\theta_k^0)$ for corresponding model predictions and each initial guess parameter θ_k^0 as follows:

$$\xi_1(\theta_k^0) = \frac{1}{R_1} \frac{\partial R_1}{\partial \theta_i^0}, \quad \xi_2(\theta_k^0) = \frac{1}{R_2} \frac{\partial R_2}{\partial \theta_i^0}. \quad (4.9)$$

▷ See chapter 3, section 3.5 for how to run these sensitivity analyses.

Sensitivity measures for model parameters

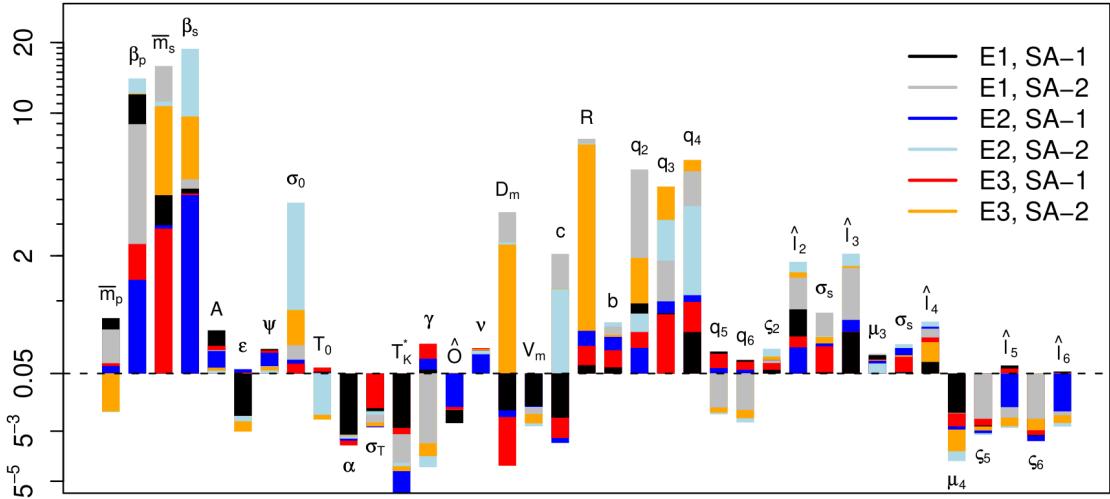


Figure 4.6: Log-scaled measures of sensitivity obtained for estimated parameters from three different experiments, E1, E2 and E3. The values below the dashed line correspond to less than 5% sensitivity of either predictions (SA-1, i.e., $\max|\xi_1, \xi_2|$) or objective function (SA-2, $\max|\xi_3, \xi_4|$) to corresponding parameter. From Senina et al. (2008).

Sensitivity of likelihood function. The second type of sensitivity analysis examines whether the objective function (which incorporates both predicted and observed data) is sensitive to model parameters. We compare values of negative log-likelihood at some found minimum $\boldsymbol{\theta}^\dagger$ to those evaluated at the boundaries of the parameter space (Vallino, 2000). We define two further measures of relative sensitivity:

$$\xi_3(\theta_k^\dagger) = \frac{L^-(\boldsymbol{\theta}^\dagger + \delta\bar{\theta}_k \cdot \mathbf{e}_k) - L^-(\boldsymbol{\theta}^\dagger)}{L^-(\boldsymbol{\theta}^\dagger)}, \quad \xi_4(\theta_k^\dagger) = \frac{L^-(\boldsymbol{\theta}^\dagger - \delta\underline{\theta}_k \cdot \mathbf{e}_k) - L^-(\boldsymbol{\theta}^\dagger)}{L^-(\boldsymbol{\theta}^\dagger)} \quad (4.10)$$

where $\delta\bar{\theta}_k = \bar{\theta}_k - \theta_k^\dagger$, $\delta\underline{\theta}_k = \theta_k^\dagger - \underline{\theta}_k$ and \mathbf{e}_k is a standard basis vector with 1 in the k th element and 0 elsewhere.

Table 4.1: Example of outputs from the sensitivity analyses.

N	parameter	relative sensitivity
1	Mp mean exp(0)	8.72181
2	Ms mean max(0)	-1.11762
3	a sst spawning(0)	2.54945 e-10
4	b sst spawning(0)	0.00656738
5	a sst habitat(0)	3.39733e-18
6	b oxy habitat(0)	-1.0208
7	MSS species(0)	-0.070417
8	nb recruitment(0)	0.0047095
9	...	
10	...	

For both -sa[] options, outputs are directly shown on the terminal screen at the end of the simulation and can be recorded in a text file if needed to produce a table (Table 4.1) or a plot (e.g., Figure 4.6). It is recommended that multiple sensitivity tests are run from experiments with different initial conditions, time series of simulation, and estimated values of parameters. This approach will help in detecting persistent patterns, for example, parameters that show consistently low or conversely really high sensitivity, and, hopefully, the reason for such patterns. If the problem cannot be solved, for example, by increasing the period of simulation to include new data or testing a better environmental forcing data set, then the parameter(s) with low sensitivity should be fixed to their best guess or estimate and removed from the optimization.

Note that local sensitivity analyses, although fast and easy to run, only provide a partial representation of observable and non-observable parameters unless the entire parameter space is explored, which is not feasible as it means exploring the entire likelihood hyper-surface in n-dimensional parametric space. Another type of sensitivity analysis, global sensitivity analysis, can be a more practical solution to analyse parameter sensitivity.

4.2.2 Global sensitivity analysis

To evaluate the model's sensitivity to its parameters, the SEAPODYM model software also includes global sensitivity analysis (GSA) based on variance methods (Saltelli et al., 2008; Pianosi et al., 2016). GSA is performed by computing the first-order ('main effect') indices measuring the direct contribution from each parameter to the output variance and the total-order ('total effect') indices measuring the overall contribution from a parameter including its interactions with other parameters. These indices are computed as follows:

$$S_i^F = \frac{\mathbf{V}_{\theta_i}[\mathbf{E}_{\theta_{\sim i}}(L^-|\theta_i)]}{\mathbf{V}(L^-)}; \quad S_i^T = \frac{\mathbf{E}_{\theta_{\sim i}}[\mathbf{V}_{\theta_i}(L^-|\theta_{\sim i})]}{\mathbf{V}(L^-)} \quad (4.11)$$

where \mathbf{E} denotes expected value and \mathbf{V} denotes variance, L^- is the negative log-likelihood (see next section), and $\theta_{\sim i}$ means varying all parameters but the i th. The main and total effect indices are useful to rank and to exclude the non-influential parameters respectively. The S_i^F measures the relative contribution of each parameter to the total output variance, $\sum_i S_i^F \leq 1$ for non-linear non-additive models. In other words, when computing first order sensitivity indices, we measure the direct contribution of each parameter (independent from the other parameters) to the output variance. The parameter θ_i is not influential if and only if the index $S_i^T = 0$. Having $S_i^T > S_i^F$ indicates an existing correlation with other parameters. Hence, total-order sensitivity indices measure the overall contribution from each parameter to the output variance, thus including interactions between correlated parameters.

According to 4.11, indices S_i^F can be computed in so-called All-At-a-Time (AAT) SA experiments, that is, randomly sampling all parameters at every model run. The evaluation of S_i^T requires One-At-a-Time (OAT) SA simulations, in which only one parameter is randomly varied in a series of model runs while others are fixed. In order to evaluate model sensitivity to its parameters given the information contained in each type of data, it is good practice to set up three SA simulation studies with model configurations integrating either catch, length frequency, or tag recapture data. As an output function, the code uses the negative log-likelihood functions that are configured for respective type of data.

First, a large number m (usually of order 10^4) of AAT simulations can be run in parallel for each data type with n parameters shuffled randomly (all at the same time) within their valid ranges, and hence each of m simulation runs has a unique subset of parameters. Second, the subset of parameters $l \ll m$ providing the lowest function values in AAT SA simulations can be selected to conduct OAT SA simulations. This allows exclusion of the model solutions that are too far from the optimum and hence unrealistic. Each OAT simulation is configured as a simulation ensemble, starting with one of the l parameter sets of length n and performing n series of runs moving iteratively from the first to the last parameter. For a given parameter, the OAT run performs k simulations with randomly sampled values within a parameter's valid range (one at a time). Note that a number $k = 25$ is chosen arbitrarily and can be easily modified in the method's computer code. In addition, the following algorithm has been found efficient to subsequently improve the cost function towards its optimum within OAT simulations, thus combining these sensitivity simulations with the likelihood profiling technique to obtain another useful diagnostic for a parameter estimation problem (section 4.4.5). The algorithm consists of starting each $i + 1$ OAT iteration with the i th parameter being fixed at a value providing the lowest function among a series of k runs. Thus, at every iteration, the OAT is moving towards lower function values, except if the subsequent parameter after being sampled k times could not provide further improvement. In this case, the initial parameter value, corresponding to the best function value in the

previous iteration, is retained. Overall, OAT simulations provide $n \times l \times k$ function values and $l \times n$ function profiles of length k .

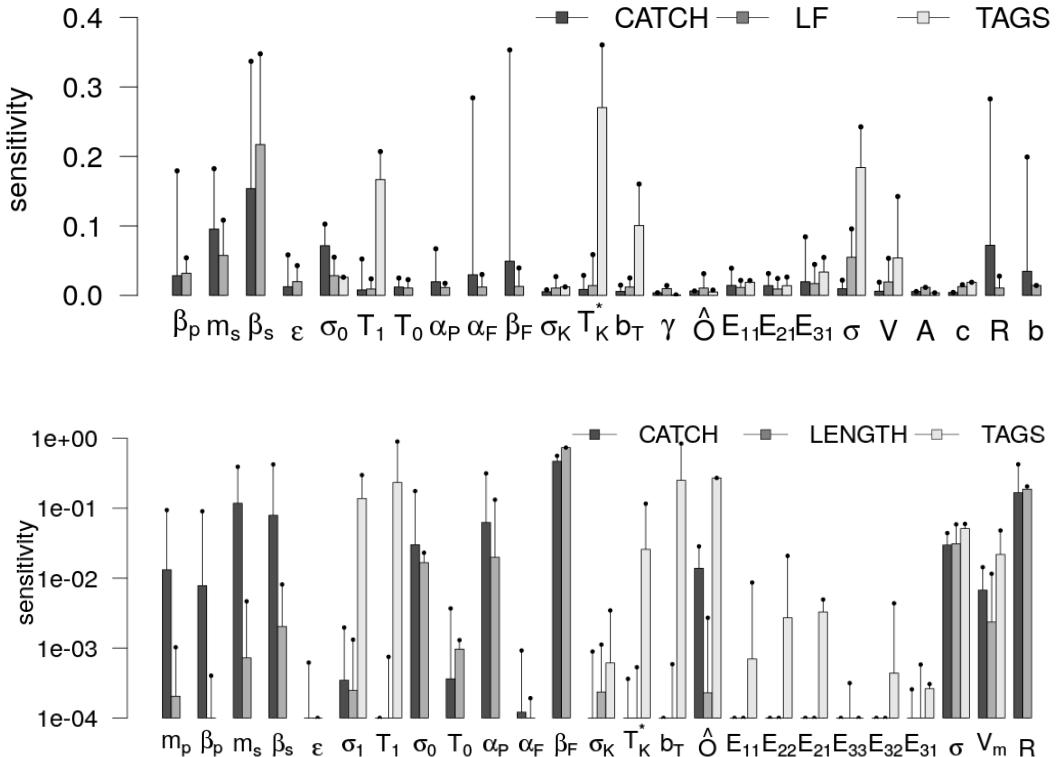


Figure 4.7: Sensitivity indices computed for three negative log-likelihood terms depending on catch, length and tagging data respectively for (top) skipjack tuna (from Senina et al., 2020b) and (bottom) bigeye tuna (from Senina et al., 2020c). Note the linear and logarithmic scales for the results in skipjack and bigeye tuna models respectively. Bars correspond to the first-order sensitivity showing direct contribution of a parameter to the model output variance, and vertical lines correspond to the total effect sensitivity, including correlations between model parameters (vertical lines).

Figure 4.7 shows the quantitative results of global sensitivity analysis (both AAT and OAT) for skipjack and bigeye tuna. Given the model and observations for these species, integrating fisheries data (catch and length frequency) into the likelihood mainly informs (provides high sensitivity to) the parameters, which influence the reproduction and mortality processes (m_p , β_p , m_s , β_s and α_p , β_F , R respectively, see Table 1.3 for description of these parameters). Interestingly, for the bigeye tuna model, the length frequency data principally

controls the estimation of recruitment, especially parameters of the spawning habitat index, α_p and β_F , that drive the seasonality of spatial distribution of recruits. But this is not the case for skipjack tuna, where the LF data provide higher sensitivity to mortality parameters than to those of recruitment. Diffusion and advection rates (σ, V_m) also have relatively high sensitivity provided by including all three types of data for bigeye tuna, but not for skipjack. For the latter species, the sensitivity to movement rates is mostly driven by tag recapture data.

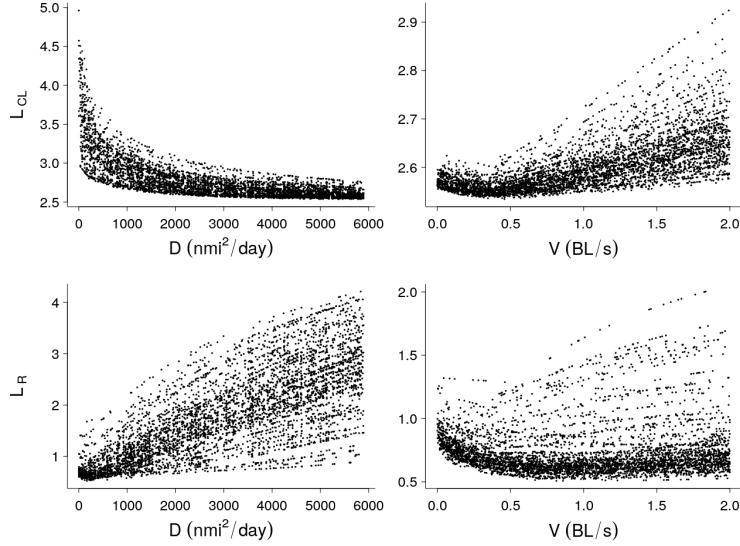
Habitat indices parameters, T_1, σ_K, T_K , and micronekton multipliers produce almost no or low change in both model outputs with fisheries data. Another habitat parameter, the dissolved oxygen threshold \hat{O} can be estimated well from the bigeye tuna catch data distributions, with its estimation probably being well constrained by strong gradients in the spatial distribution of catches around zones with low oxygen content in the upper mesopelagic layer. For skipjack tuna, primarily occupying the epipelagic layer, which has mostly high oxygen levels, there is a weaker signal for the estimation of this parameter. These two results clearly demonstrate that sensitivity to model parameters is driven by data, their spatio-temporal coverage and the signal they provide to constrain model parameters.

Likelihood profiles for movement parameters, obtained with help of OAT simulations with skipjack and bigeye tuna models are shown in Fig. 4.8. The presence of shape in profiles indicates that the model is sensitive to a given parameter. And the opposite, the flatness and dispersion of profiles, suggests that this parameter is not informed by the data. Profiles for movement rates in both skipjack and bigeye tuna models clearly show the tendency of fisheries data to increase diffusivity and reduce directed movement rates, both contributing to non-zero biomass everywhere in the model domain and low patchiness. The overall pattern is similar in the two models, although there is a slightly increasing trend in length frequency likelihood with increasing values of diffusion rate in the bigeye model and a better defined shape of tagging data likelihood with respect to advection rate, with a minimum located between 0.5 BL/s and 1 BL/s. Overall, integration of tagging data in the models of skipjack and bigeye tuna populations enhances observability of movement parameters while modifying the shape of the likelihood hyper-surface with minimal values towards lower diffusions and non-zero advection rates. See more results and discussion on likelihood profiles in skipjack and bigeye tuna models in recent SEAPODYM studies (Senina et al., 2020b,c, 2021).

4.3 Maximal likelihood estimation method

To describe spatio-temporal dynamics of fish populations, SEAPODYM relies on parameters that describe the population dynamic rates and provide links between environmental variability and intrinsic dynamic processes such as reproduction, survival and movement. In order to have confidence in model predictions, particularly in the adequacy of the population dynamical responses to environmental variability, we need to combine simulations with quantitative parameter estimation. In other words, we need to inform model parameters,

a) skipjack tuna



b) bigeye tuna

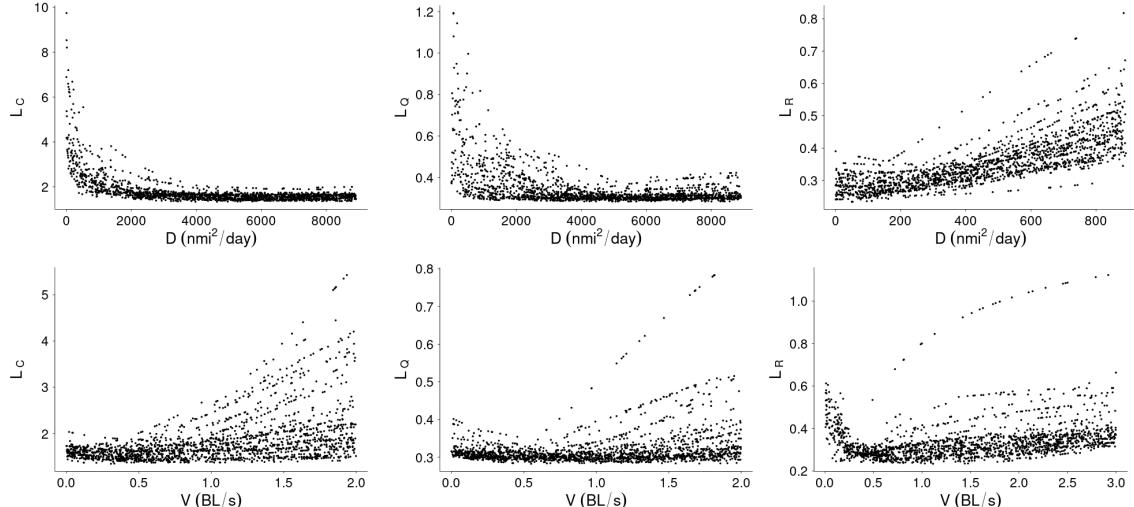


Figure 4.8: Likelihood profiles, obtained as a result of OAT simulations for skipjack and bigeye tuna. The scatterplots for the skipjack model correspond to two sets of simulations set up for combined catch and length frequency likelihood and for tagging data likelihood. (From Senina et al., 2020b). For bigeye tuna, each data likelihood term was studied separately (same results are shown in Senina et al., 2020c). Note, parameter D values correspond to the highest diffusion rate of largest tunas in the population if habitat is null and parameter V is the velocity of density field (in body lengths per second) at highest habitat gradient, that is $\nabla H = 1$.

$\boldsymbol{\theta}$, by fitting the model predictions to the observational data. This is done via use of the maximum likelihood estimation (MLE) method.

The first implementation of MLE in SEAPODYM is described in Senina et al. (2008). The likelihood function including a new tagging data term is constructed of three observed variables: geo-referenced catch C_{fKij} by fishery f , for the time step K and in the observational grid cell IJ ; length frequency distributions Q_{fKr} in the sampling zone r , and the spatial distributions of tag returns $R_{qI'J'}$ in quarter q and the grid cell $I'J'$ of resolution defined for the prediction of tag recaptures. The complete likelihood function of unknown parameters $\boldsymbol{\theta}$ is computed as follows:

$$L = \prod_{fKIJ} (L(\boldsymbol{\theta} | C_{fKij})) \cdot \prod_{fKr} (L(\boldsymbol{\theta} | Q_{fKr})) \cdot \prod_{qI'J'} (L(\boldsymbol{\theta}_{\text{tm}} | R_{qI'J'})) , \quad (4.12)$$

where parameters $\boldsymbol{\theta}_{\text{tm}}$ are the subset of parameters of model 1.1, which control the dynamics of tag model 4.5. The MLE is obtained by solving the following constrained minimization problem:

$$\boldsymbol{\theta}_{\text{mle}} = \underset{\boldsymbol{\theta} \in (\boldsymbol{\theta}, \bar{\boldsymbol{\theta}})}{\operatorname{argmin}} (L^-) \quad (4.13)$$

for $L^- = -\ln L$. The problem 4.3 is solved iteratively by a quasi-Newton method until the conditions defining a minimum are fulfilled, namely $\frac{\partial(L^-)}{\partial \boldsymbol{\theta}} \simeq 0$ and the Hessian matrix is positive definite \mathbf{H}_{++} .

4.3.1 Adjoint computer code

Solving the problem 4.3 of function minimization with a gradient method such as a quasi-Newton requires evaluation of the derivatives of the function with respect to each control parameter. In addition, the gradient of the negative log-likelihood, $\frac{\partial(L^-)}{\partial \boldsymbol{\theta}}$, must be computed to the same accuracy as the L^- . In SEAPODYM, we evaluate the function derivatives with help of adjoint computer code, also called reverse mode differentiation code (see e.g. Griewank and Corliss (1991)). Direct and derivative code of SEAPODYM is implemented using the objects and memory control of AUTODIF library (Autodif User's Manual, 2021), created by David Fournier (Otter Research Ltd, 1994) for automatic code differentiation. This library also provides the quasi-Newton function minimizer and parameter scaling algorithm for solving the constrained optimisation problem (see below). The use of AUTODIF and its capacity for automatic code differentiation facilitates model developments; however, writing adjoint computer code in order to increase efficiency of computations is always recommended. Thus, the runtime and the memory use can be significantly improved by including appropriate recalculations of the forward model within the adjoint code, rather than storing all the intermediate variables in computer memory.

When writing new or modifying existing adjoint code, because of the change in the forward code, the resulting analytic derivatives must be verified. This can be done using the AUTODIF derivative check, which uses the central finite difference as the derivative approximation:

$$\frac{L^-(\theta_k + h) - L^-(\theta_k - h)}{2h} - \nabla_k L^- = O(h^2),$$

that is the discrepancy between each gradient component $\nabla_k L^-$ obtained by analytic differentiation and its finite difference approximation changes parabolically with step h , varied from 10^{-6} to 10^{-1} .

```

Control variables in optimisation:
1. b sst habitat(0) = 13.4268
2. T age size slope(0) = 1.06455
3. eF habitat epi(0) = 0.745657
4. eF habitat mmeso(0) = 0.910203
5. eF habitat mbathy(0) = 0.295928
6. MSS species(0) = 0.469338

Initial statistics: 6 variables; iteration 0; function evaluation 0
Function value 1.1059255e+03; maximum gradient component mag 4.1510e+02
Var Value Gradient |Var Value Gradient |Var Value Gradient
 1 0.23587 -1.75905e+02 | 2 -0.77001 -2.62749e+02 | 3 0.32697 1.38282e+02
 4 0.61250 2.58367e+02 | 5 -0.26765 4.15103e+02 | 6 -0.45447 2.20853e+02
...
quasi-Newton iterations

...
- final statistics:
6 variables; iteration 16; function evaluation 28
Function value 8.8558e+02; maximum gradient component mag 9.8197e-01
Exit code = 1; converg criter 1.0000e+00
Var Value Gradient |Var Value Gradient |Var Value Gradient
 1 0.78651 3.56932e-01 | 2 -0.57958 3.91102e-01 | 3 -0.05783 -1.08384e-01
 4 0.43575 -3.57419e-01 | 5 -1.68728 9.45632e-01 | 6 -0.58525 9.81975e-01

Estimated parameters:
1. b sst habitat(0) = 15.319
2. T age size slope(0) = 1.21025
3. eF habitat epi(0) = 0.454642
4. eF habitat mmeso(0) = 0.816134
5. eF habitat mbathy(0) = 0.264148
6. MSS species(0) = 0.339454

```

Figure 4.9: Extract from an example optimization run showing the initial and estimated parameters θ_k and the optimization statistics tables: for each parameter the table provides the value of rescaled parameter θ'_k and corresponding derivative of the likelihood function.

4.3.2 Parameter scaling

Parameter scaling is done to keep parameters within the allowed range, defined as parameter boundaries. Instead of setting penalties to the boundaries of θ , the AUTODIF library performs a constrained minimization through parameter scaling (see e.g., Bard, 1974; Vallino, 2000). The latter implies that the optimization routine operates in the unbounded parametric space that is mapped to the bounded one with the transformation:

$$\theta_k = \underline{\theta}_k + (\bar{\theta}_k - \underline{\theta}_k) \left(1 + \sin \frac{\pi \theta'_k}{2} \right), \quad (4.14)$$

that is, the variable parameter θ'_k can vary from $-\infty$ to ∞ while θ_k remains within the imposed bounds. So, parameter scaling is done prior to optimization, which operates θ'_k obtained from eq. 4.14 parameters, and at the end of optimization the parameters are rescaled back to θ_k (see Figure 4.9).

4.3.3 Likelihood functions in SEAPODYM

The negative log-likelihood, $L^- = -\ln(L)$ to be minimized is the sum of three components: L_C^- , the catch data; L_Q^- , the length-frequencies data; and L_R^- , the tagging data component. These components are described below. For simplicity, let us use notations f, a, t, i, j, r for fishery, age, time, longitudinal, latitudinal and regional indices respectively, knowing that they correspond to data-specific strata defined for each type of observations as described in sections 4.1.1–4.1.3.

Catch and CPUE likelihoods

Several functions are implemented in SEAPODYM to account for different statistical distributions of the catch data and hence to define likelihood function components. One frequently used assumption is that observing a non-zero catch event is a rare event and that these observations are described by a Poisson distribution. In this case the likelihood function is equal to the Poisson probability mass function, that is:

$$L(\theta | C^{obs}) = \prod_{tfij} \frac{C_{tfij}^{pred} C_{tfij}^{obs} e^{-C_{tfij}^{pred}}}{C_{tfij}^{obs}!}, \quad (4.15)$$

and the corresponding negative log-likelihood is therefore

$$L_C^- = \sum_{tfij} \left(C_{tfij}^{pred} - C_{tfij}^{obs} \ln(C_{tfij}^{pred}) + \ln \Gamma(C_{tfij}^{obs} + 1) \right). \quad (4.16)$$

The Poisson likelihood is mostly used for the purse-seine fleets targeting the modelled species (see e.g., Senina et al., 2008). However, this function cannot be used when fishing

data contain many zeros, which is often observed for the species that constitutes an accidental catch or is a secondary target for the fleet. In this case, it may be more appropriate to use other distributions, for example, a negative binomial distribution with zero inflation:

$$L(\Theta|C^{obs}) = \begin{cases} \prod_{tfij} \left(p_f + (1 - p_f) \left(\frac{\beta_f}{1+\beta_f} \right)^{\frac{\beta_f C_{tfij}^{pred}}{1-p_f}} \right), & \text{if } C_{tfij}^{obs} = 0, \\ \prod_{tfij} \left((1 - p_f) \frac{\Gamma\left(C_{tfij}^{obs} + \frac{\beta_f C_{tfij}^{pred}}{1-p_f}\right)}{\Gamma\left(\frac{\beta_f C_{tfij}^{pred}}{1-p_f}\right) C_{tfij}^{obs}!} \left(\frac{\beta_f}{1+\beta_f} \right)^{\frac{\beta_f C_{tfij}^{pred}}{1-p_f}} \left(\frac{1}{1+\beta_f} \right)^{C_{tfij}^{obs}} \right), & \text{if } C_{tfij}^{obs} > 0, \end{cases} \quad (4.17)$$

where the parameters β_f and p_f , are the negative binomial parameters, the former showing how much variance exceeds the expected value and second is the probability of getting a null observation. Both are estimated in the optimization process. Correspondingly, the negative log-likelihoods are

$$\bar{L}_C = \begin{cases} \sum_{tfij} \left(-\ln \left(p_f + (1 - p_f) \left(\frac{\beta_f}{1+\beta_f} \right)^{\beta_f \mu_f} \right) \right), & \text{if } C_{tfij}^{obs} = 0, \\ \sum_{tfij} \left(\ln \Gamma(\beta_f \mu_f) + \ln \Gamma(C_{tfij}^{obs} + 1) + \ln(\beta_f + 1) (\beta_f \mu_f + C_{tfij}^{obs}) - \ln(1 - p_f) - \ln \Gamma(\beta_f \mu_f + C_{tfij}^{obs}) - \beta_f \mu_f \ln(\beta_f) \right), & \text{if } C_{tfij}^{obs} > 0, \end{cases} \quad (4.18)$$

where $\mu_f = \frac{C_{tfij}^{pred}}{1-p_f}$. Other available likelihood functions, implemented in SEAPODYM for catch or CPUE data, are derived from truncated Poisson, negative binomial, exponential, Weibull, and log-normal probability distributions.

Also, if the normal distribution assumption can be made about the catch or CPUE data, then the negative log-likelihood function can be defined as the simple least-square function or as the negative concentrated log-likelihood that is equivalent to the negative log-likelihood for a normally distributed random variable with minimal variance (Otter Research Ltd, 1994). For example, for the fisheries, for which the catch removal method is used, we choose the normal likelihood profile because the errors are proportional to the modelled biomass and hence can be assumed normally distributed:

$$\bar{L}_C = \frac{1}{2\sigma_C^2} \sum_{f,t,i,j} (C_{ftij}^{pred} - C_{ftij}^{obs})^2.$$

Note that there is not much sense in converting catch predicted with the catch removal method to CPUE, as in this case predicted catch does not depend on fishing effort. Also, the use of observed catch in the catch removal approach may result in $L_C^- = 0$ for all fij if there is always enough biomass to support the observed catch (see eq. 4.3). Therefore, another likelihood term, allowing the biomass constraint, is necessary (see section 4.3.3).

Length frequency likelihoods

Note that proportions of catch-at-age given by eq. 4.4 do not depend on fishery-specific catchability coefficients. Hence, using only LF data in the likelihood, these parameters cannot be estimated. Two functional forms of likelihoods for LF data are implemented in SEAPODYM, both based on the assumption of normal distribution of the small-scale LF observations. The first function gives the following contribution from length frequency data to the negative log-likelihood:

$$L_Q^- = \frac{1}{2\sigma_Q^2} \sum_{t=1}^{tm} \sum_{f=1}^{n_f} \sum_{a=1}^{n_a} \sum_{r=1}^{n_r} (Q_{fatr}^{pred} - Q_{fatr}^{obs})^2, \quad (4.19)$$

where the proportion at age a in the catch computed from the observed length frequency data is $Q_{fatr}^{obs} = \frac{N_{fltr}^{obs}}{\sum_l N_{fltr}}$, $l \in [\underline{l}_a, \bar{l}_a]$. N_{fltr}^{obs} is the number of fish of length l in region r that belongs to the cohort of age a . The variance σ_Q^2 is assumed to be a constant value set up in the computer code, which cannot be estimated.

The second type of LF likelihood function uses the formulation proposed by Hampton and Fournier (2001) for the fitting to the length frequency data. It is called the robustified likelihood:

$$L_Q^- = 0.5 \sum_{f,t,r} \log \left(2\pi \left(\xi_{ftr} + \frac{1}{I} \right) \right) + I \sum_{f,t} \log(\tau_{ft}) + \sum_{f,a,t,r} \frac{(Q_{fatr}^{obs} - Q_{fatr}^{pred})^2}{2\tau_{ft}^2 \left(\xi_{ftr} + \frac{1}{I} \right)}, \quad (4.20)$$

where the constant $\xi_{ftr} = Q_{ftr}^{obs}(1 - Q_{ftr}^{obs})$, $\tau_{ft} = \frac{P}{\min(1000, S_{ft})}$ and I is the number of size intervals in the original samples (before redistribution to the model age structure). The term $\min(1000, S_{ft})$ embodies an assumption on the maximal sample size to be considered accurate and the constant P assigns how much the variance of the LF sample is greater than that of the truly random sample of a given size. See more details on this likelihood in Hampton and Fournier (2001).

Tag recapture likelihood

The contribution to the negative log-likelihood function from the tag recaptures data can be defined based on the assumption of the number of tag recaptures being normally distributed:

$$\bar{L}_R = w_R \sum_{q,I,J} \left(R_{qIJ}^{pred} - R_{qIJ}^{obs} \right)^2, \quad (4.21)$$

where w_R is a constant allowing equal weighting between the different likelihood terms, R_{qIJ}^{pred} and R_{qIJ}^{obs} are the number of tag recaptures predicted and observed respectively in quarter q and coarse resolution cells IJ . Alternatively, the log-normal likelihood can be defined based on the assumption that tag recapture data are log-normally distributed. Since the tag recapture data may contain true zeroes, they can be accounted for by adding a small constant to both predictions and observations, hence leading to the following contribution to the negative log-likelihood:

$$\bar{L}_R = w_R \sum_{q,I,J} \left(\ln \left(R_{qIJ}^{pred} + c \right) - \ln \left(R_{qIJ}^{obs} + c \right) \right)^2, \quad (4.22)$$

where the constant $c \ll 1$ is currently set up directly in the computer code.

Biomass constraint

Another objective related to the use of catch removal method is to estimate the minimal population stock with an age and spatial distribution that would support the observed catches given the selectivity of fishing gears. By definition in this method, predicted catch in the grid cell is exactly the observed catch if the modelled biomass is sufficient to sustain it (eq. 4.3). In this case the observed catch is simply subtracted from the biomass, and the predicted catch is equal to the observed. Therefore, predicting biomass above the observed catch in all strata is optimal with this method, but increasing the biomass to achieve zero error may lead to an overall model solution that is totally unrealistic. On the other hand, the use of such predictions in the optimization provides the observability of model parameters from the grid cells where the predicted biomass is lower than the observed catch. In this case, the predicted catch is equal to the available biomass, the biomass becomes zero after catch removal and the contribution to the likelihood becomes non-zero. In order to prevent an overestimation of the biomass and to facilitate the convergence towards a realistic solution with the minimal biomass supporting the observed catch, the likelihood must include one additional term allowing the minimization of the total biomass to the lowest possible levels. Fitting the predicted to the observed catch then means reducing the local discrepancies in the case of the lack of biomass. This can be achieved by adding the following contribution to the negative log-likelihood:

$$\beta = \left(\frac{1}{n_t} \sum_{atij} (N_{atij} W_a \Delta x_i \Delta y_j) - \bar{B} \right)^2, \quad (4.23)$$

where n_t is the number of model time steps in the likelihood computation, W_a is the mean fish weight at age a and \bar{B} is the mean stock biomass, assumed to be minimal for the whole domain or a specific part of it. Hence, in the case of using fisheries accounted with observed catch only, the function to be minimized becomes $L^- = -\ln L + \beta$, where L includes at least two other likelihood terms, for example, effort-based catch for selected fisheries and LF likelihoods, or the LF and the tagging data likelihoods. Given the poor observability of model parameters from catch data (Figure 4.8), it is essential to use data with good spatial and species life history coverage to estimate those reproduction, mortality, habitat and movement parameters that control spatial distributions of population density at different ages.

The biomass \bar{B} can be chosen empirically in a series of optimization experiments so that at any time step and grid cell the exploited biomass $\sum_{af} (s_{af} N_{tij} W_a \Delta x_i \Delta y_j) \geq 0.8 C_{tij}^{obs}$, where s_{af} is the fisheries selectivity. Ideally the level of biomass should always be higher than the observed catch; however, up to 20% local errors can be allowed because of biases in the physical forcing, the errors in the fishing data and the coarse spatial resolution of the numerical model.

4.4 Parameter estimation workflow

The search for a best set of parameters, corresponding to the global minimum is not a trivial task within complex non-linear models, and requires lots of optimization experiments that differ in the configuration of model structure, the data to be used in the likelihoods, likelihood definitions, the set of variable and fixed parameters and finally the initial parameter values. Here are some brief statistics of optimizations undertaken to achieve the MLE solutions described in Senina et al. (2020b) with catch and length (CL) and catch, length frequency and tag recapture data (CLT) for skipjack tuna. About 40 optimization experiments with CLT likelihood, corresponding to $\approx 15,000$ function evaluations (FE) in total, 25 optimizations for CL (≈ 6000 FE) and 310 optimizations (≈ 150 FE by optimization on average) with tagging likelihood only. Every parametrisation obtained as a result of function minimization, has to be analysed, the parameter estimates confronted with the best available knowledge about the modelled species, and the fit and errors and biases evaluated.

This section describes the parameter estimation workflow, covering some aspects of the optimization experiment set-up and available tools for troubleshooting and additional model and method analyses in case if the method does not converge or if the MLE solution is not satisfactory.

4.4.1 Problem dimensionality and memory control

First, to maintain the time of computation at a reasonable level, it is necessary to find a balance between the number of observations, the size of the domain, the spatial and temporal

resolution, and the extension of the time series used in the optimization experiment. The dimensionality of the optimization problem affects the amount of memory needed to store all intermediate variables for adjoint calculations.

The AUTODIF library controls memory allocation. Three buffers are declared with their sizes defining the memory for the variables dependent on parameters θ , called *dvariables*, and for their derivatives. The names of these buffers in SEAPODYM are: `gs_var_buffer` – allocated to store *dvariables* and all AUTODIF objects declared in the code, `cmpdiff_buffer` – to store the gradient information as well as intermediate variables used by the adjoint code for computing derivatives, and `gradstack_buffer` – to store the information necessary to calculate derivatives via automatic differentiation. Note that since the adjoint functions are written in SEAPODYM to compute analytical derivatives, the third buffer is usually kept small.

It is important that the size of the buffers is set up as large as required to store all information in the physical memory during the optimization runs. Otherwise, if the size of the buffer is insufficient, the program will write its contents to the hard disk, into files `cmpdiff.tmp`, `gradfil1.tmp` and `gradfil2.tmp`, which will considerably slow down the program execution. The indicator of correct configuration of the gradient structure buffers is that these temporary files have size 0. The amount of memory is declared in the program code, in function `main`, by setting the buffer size in the global variables and while declaring a gradient structure object:

```
gradient_structure::set_GRADSTACK_BUFFER_SIZE(gradstack_buffer);
gradient_structure::set_CMPDIF_BUFFER_SIZE(cmpdif_buffer);
gradient_structure gs(gs_var_buffer).
```

See the AUTODIF documentation (Autodif User’s Manual, 2021) for more details on the format and the estimation of these buffer sizes.

During optimization, the SEAPODYM software writes the optimization report file in the run directory named ‘`optim.rep`’. It is updated upon each successful iteration. This file contains the intermediate parameter values (without scaling), the function value and the maximal gradient as well as the execution time per function evaluation. This file is particularly useful for estimating the total runtime of the optimization experiment. The rule of thumb is to multiply the number of parameters by three to get the total number of iterations.

4.4.2 Likelihood configuration

The available likelihood configuration parameters are all listed in Chapter 3, section 3.4.6. The choice of the likelihood terms is done by setting the flags `frq_likelihood`, adding or removing the term 4.20, `tag_likelihood`, adding or removing the term 4.21, and the flag `stock_likelihood` to control the term 4.23. Note that catch likelihoods (defined by fishery)

are always active and only the types of these likelihoods can be selected in the configuration parfile. However, it is possible to exclude certain fisheries from the likelihood computation via modification of vector `mask_fishery_likelihood`. The type of catch variable, that is the use of catch or CPUE in the likelihood, can be set up via flag `like_c_cpue`. The average stock value as well as the region over which this value is computed can be configured in the parfile (see `mean_stock_obs` and corresponding regional parameters).

Finally, it is possible to solve optimization problem with tagging data only. This is activated by the flag `tag_likelihood_only`. In the latter case only model 4.5 will be solved for the tagged cohorts defined in the input tagging data files, which are also provided in the parameter file (see Appendix A.2).

4.4.3 Initial conditions

We distinguish between the initial conditions of optimization, that is the initial parameter values and the initial conditions of model 1.1, that is the model's initial state vector.

Initial parameter values

It is important to configure and run multiple optimization experiments starting from different initial parameter values. This method uses what are called ‘jitter’ optimization runs. The approach of perturbing the model parameters and restarting the experiment prevents the model converging towards the local minima, although it is not a rigorous method to ensure that the found solution is a global minimum. For the latter, it is necessary to use the global optimization methods such as simulated annealing (Matear, 1995) or a genetic algorithm.

Initial state vector

Regarding the initial distributions of population density, they may also greatly influence parameter estimation as the model numerical solution depends on them. The sensitivity to the initial conditions can be evaluated by running several optimization experiments starting from different initial states. The influence of initial conditions on the results of minimization can be reduced by skipping the first predictions from being used in the likelihood computation. The number of time steps to skip is configured by the parameter `nb_step_to_skip` in the parfile. The simplest way is to exclude the number of time steps corresponding to the age at 50% maturity of modelled species, that is the time necessary for the new larval production to enter the mature stock and the mature stock to be redistributed according to the currently estimated parameters driving the spatial dynamics. Note that it might be impractical to skip a long period and hence a large number of observations in the case of long living species with late maturity. Thus, realistic assumptions on the population abundance and availability of spatial distributions from previous optimizations might be necessary.

4.4.4 Role of environmental forcing

The parametrization of SEAPODYM is highly sensitive to ocean forcing. Therefore, a realistic ocean forcing, that is, based on observations, is necessary to drive the population dynamics. Realistic physical and biogeochemical variables are necessary to explain the observed variability within the large datasets of tuna catches and length distributions, which in turn reflect changes in fish abundance due to both fishing and the impact of environmental variability, including ENSO (El Nino southern oscillation) events.

Constraining the population dynamics of SEAPODYM, environmental data can be viewed as ‘fixed’ parameters, and it is essential that these parameters are ‘optimal’. However, environmental forcing is itself modelled data and has biases. In addition, optimization experiments are usually configured with the coarse spatial resolution model, which is characterized by weak representation of ocean circulation and subsequent biases in primary production and micronekton densities. For example, the correct temperature variability is of primary importance to determine seasonal spawning or feeding migrations. A model that is unable to adequately describe the seasonal migrations because the temperature seasonality and spatial structure have biases in the geographic area being the population spawning or feeding ground or the area of migration route, will tend to bias parameters of habitat indices and to extend the overall biomass distribution, if data from a corresponding geographic area will be used in function minimization. Consequently, it is advisable to exclude from the likelihood calculation the data associated with the complex current systems if the forcing data underestimates them, or the data showing highly heterogeneous patterns if the model spatial resolution does not allow reproducing them. Note that excluding fisheries from the likelihood function does not mean excluding them from the computation of fishing mortality. However, each case should be examined in the context of its impact on both the maximum likelihood estimation and the local mortality rates.

4.4.5 Two-dimensional projection of likelihood function

Sometimes it is useful to explore the likelihood function visually, plotting different projections of likelihood hyperspace over the pair of selected parameters. This may give a clue about the observability of a certain parameter, whether the boundaries of the parameter should be modified, how the penalty function changes the shape of the likelihood, and so on. Plotting the cost function is also a simple way to visualize the localisation of the solution found. For example, the likelihood function can have multiple minima and the likelihood projection plot can reveal the local minimum problem and help to navigate the minimization experiments further. The numerical instability problems, or the high-mode non-linearity of model solutions, and/or noisy observations can result in abrupt, spiky surfaces of the cost function and hence impede the convergence of the minimization method. Figure 4.10 illustrates an example of the the cost function having two local minima, with one local minimum around 10°C for preferred habitat temperature, which is totally unrealistic for yellowfin tuna.

The two-dimensional likelihood profiling is activated by execution option ‘-p’ (see Chapter 3, section 3.5.5 for details on configuring and running these simulations).

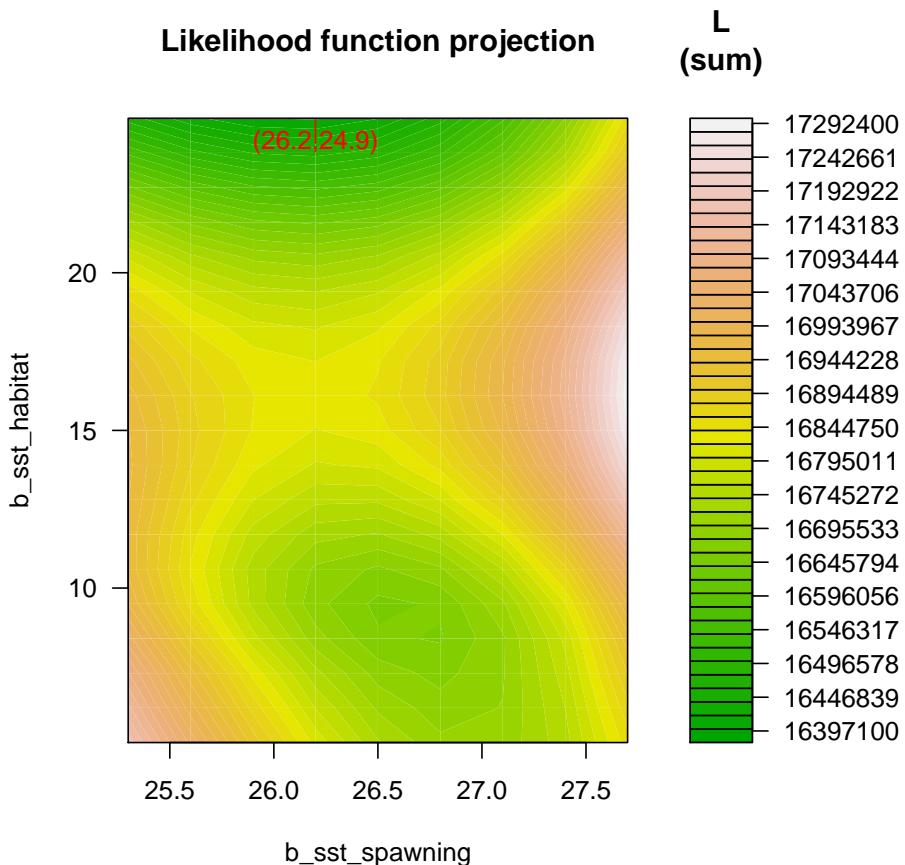


Figure 4.10: Likelihood projection over two habitat parameters in the yellowfin tuna model.

4.4.6 Identical twin experiments

Demonstrating that a solution found by numerical optimization is a global solution and not a local minimum is a difficult and well-known problem in all non-linear optimization problems and data assimilation. Evaluation of the uniqueness of the estimated parameters is challenging because of the complexity of the ecosystem model, the high dimension of the objective function and scarcity of available observations (Robinson and Lermusiaux, 2002; Vallino, 2000). Unfortunately, one cannot prove that minima found by the gradient descend method are not local minima. Considering the computer time to perform one experiment, it is also unrealistic to envisage an exhaustive study that would allow us to conclude that computed solutions are global. It is possible, however, to evaluate the reliability of the

solutions by conducting so-called ‘identical twin experiments’. They can verify that both the model and the method allowed the estimation of chosen parameters using the available number of observations. These tests consist of estimating parameters from artificial data series constructed from predictions given by the model. If optimization works well with the model and experiment set-up, then after sufficient perturbation of optimal parameters we should be able to retrieve them, because they determine a known a priori solution represented in the artificial data series.

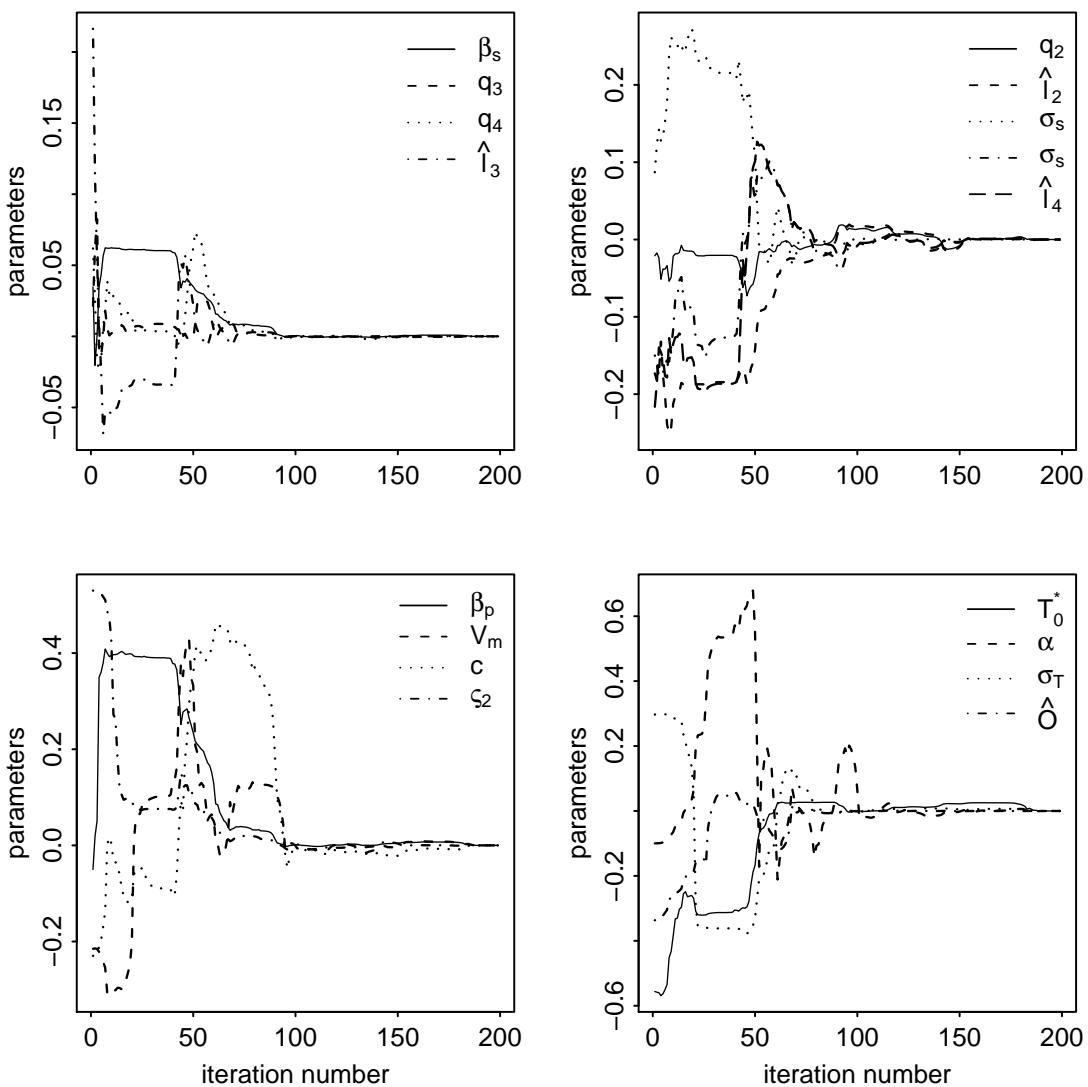


Figure 4.11: Evolution of control parameters during twin data experiment conducted for the artificial data simulated with skipjack tuna model configuration. Parameters are grouped by their sensitivities in descending order. (From Senina et al., 2008).

Configuring and running the twin experiments with SEAPODYM can be done in the following steps:

- run a simulation with selected set of parameters;
- using the model predictions for catch, length and tag recapture data, create an artificial data set, with or without adding noise to the data, format the model outputs as the input data;
- perturb the initial values of the parameters to be estimated in the parameter file;
- run SEAPODYM in the optimization mode until convergence to the MLE solution;
- use the optim.rep output file to check the result and produce plots.

Twin experiments conducted for the skipjack tuna application (Senina et al., 2008) have shown that the control parameters were successfully recovered with small relative errors $\varepsilon < 0.001$ due to the computer round-off error. The evolution of the parameters during the minimization process can be plotted (see figure 4.11) to illustrate how quickly the parameters were recovered, and which parameters are the most difficult to correctly estimate.

4.4.7 Correlations between parameters

To be well determined, the model parameters, i.e. control variables in optimization, must be independent. However, correlation between model parameters is a common issue. For example, increase of the larval recruitment rate on one side and increase of the predation mortality of larvae on the other can provide the same number of recruits, and hence the simultaneous estimation of these two parameters can lead to biased estimates of the total population size. If parameters are correlated, the minimization procedure will increase the number of iterations, leading to an overall increase in the computational time. Establishing correlations between all model parameters can help reduce the number of parameters to enable their successful estimation, while fixing their correlated pairs at reasonable and meaningful values.

Correlation coefficients between pairs of estimated parameters can be derived from the off-diagonal elements of the error-covariance matrix, which is the inverse of the second-order derivative, or Hessian, matrix:

$$\mathbf{C} = \mathbf{H}^{-1}, \text{ with } \mathbf{H} = \frac{\partial^2 L^-}{\partial \theta_i \partial \theta_j}, \quad i, j = 1, 2, \dots, n \quad (4.24)$$

that the model has converged towards the minimum of the experiment. Then, to get the parameter estimation error, we compute the variance of the estimated parameters from the error-covariance matrix 4.24 obtained from the Hessian matrix. Note that the Hessian matrix \mathbf{H} must be evaluated with $\boldsymbol{\theta}_{\text{mle}}$, that is at the minimum of the negative log-likelihood function.

The diagonal elements of the variance-covariance matrix \mathbf{C} provide estimates of the variance of the optimal parameters. From the variance we derive the standard deviations, providing the measure of uncertainty on the parameters estimates. Small uncertainty is an indicator that the parameter was well estimated from observational data given the found minimum. Large uncertainties obtained in this error analysis are good indicators of poorly determined (poorly observed) parameters.

4.6 Model validation

The validation of model solutions can be done via the following steps:

- evaluation of the improvement of the fit to the data due to optimization;
- analysis of model dynamics under estimated parameter values;
- measuring the fit with the optimal solution to an independent dataset.

To measure how well the model describes the data used in optimization and how well it fits to the independent data, we can use three statistical metrics: (1) the coefficient of determination r^2 (squared Pearson correlation coefficient), reflecting the percentage of predicted variability that is consistent with observations; (2) the standard deviation ratio σ_{rel} , that is the ratio between standard deviations of model predictions and those of data; and (3) the normalised mean square error, NMSE (eqs. 4.25–4.27). The best score is 1 for r^2 and SDR, and 0 for NMSE. Note, these metrics can be computed and compared between different model predictions for: (1) R – tag recaptures, (2) C – catch predicted by the catch removal method, and (3) EC – catch based on the fishing effort. Note that to compute the scores for EC , the catchability parameters must be estimated for those fisheries that are used in optimization with the catch removal method.

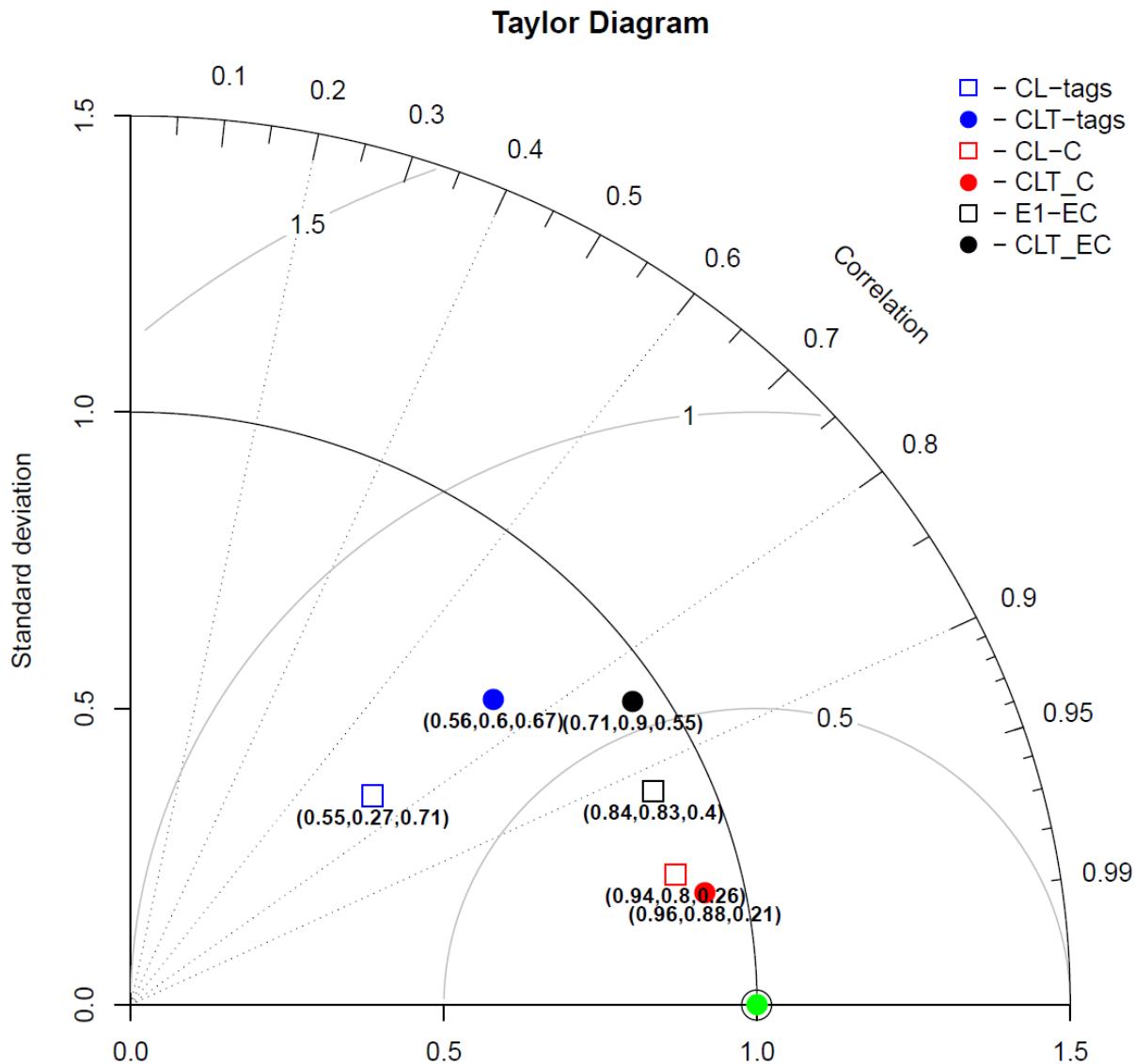


Figure 4.13: Taylor diagram providing three aggregated validation metrics of model fit to the data: 1) correlation shown in angular coordinates, 2) standard deviation ratio $\frac{\sigma_{pred}}{\sigma_{obs}}$, shown as a distance from (0,0) point, that is depicted on both, horizontal and vertical axes, and 3) the normalized mean squared error shown as a distance from the best score (1,1,0) depicted by concentric circles. Each filled or empty shape on the Taylor diagram corresponds to three metrics corresponding to the fit by a given model (letters before dash) to a given data (letters after dash): CL (same as E1) denotes the MLE model obtained with catch and length frequency data only, CLT - MLE model with catch, length frequency and tag recapture data, C and EC mean that the fit is measured for catch predicted with catch removal and effort-based methods respectively (From Senina et al., 2020b).

Let x denote the set of observations and y the model predictions of length N . Statistical metrics are computed and plotted on a Taylor (Taylor, 2001) diagram (Fig 4.13) as follows:

$$r^2 = \frac{\left(\frac{1}{N} \sum_{i=1}^N (x - \bar{x})(y - \bar{y}) \right)^2}{\sigma_x^2 \sigma_y^2} \quad (4.25)$$

$$SDR = \sigma_y / \sigma_x \quad (4.26)$$

$$NMSE = \frac{\left(\frac{1}{N} \sum_{i=1}^N ((x - \bar{x}) - (y - \bar{y}))^2 \right)^{\frac{1}{2}}}{\sigma_x} \quad (4.27)$$

where σ_x and σ_y are the standard deviations of observations and predictions respectively, \bar{x} and \bar{y} are the means, and NMSE is the centred root-mean-square error normalized by the standard deviation of the observations.

Note that the independent data are the data that were not used in the likelihood. We can extend the data time series simply by including several years of data left out for validation, and then to compute the validation scores. However, it is often hard to leave enough data to validate a highly-dimensional model. The best validation of the model with spatial dynamics is its application to a large independent set of data both in time and space and in a comparison with the baseline. An example illustrating this method is described in the recent SEAPODYM study (Senina et al., 2020a). The MLE solution obtained for the South Pacific albacore population was applied to simulate the dynamics of the Atlantic stock. Taking into account that the model with optimization performs well on the data that were used in the objective function and usually outperforms the model that was specialized on a different set of data, the MLE was also obtained using the Atlantic Ocean fisheries data. The validity of the South Pacific model was then demonstrated by comparing the statistical metrics computed for its predictions for the Atlantic Ocean population with those computed for the predictions of the Atlantic (baseline) MLE model.

References

AUTODIF: A C ++ Array Language Extension with Automatic Differentiation For Use in Nonlinear Modeling and Statistics. <https://github.com/admb-project/admb/releases/download/admb-12.3/autodif-12.3.pdf>

Bard, Y. Nonlinear parameter estimation. Academic Press: New York, 1974.

Griewank, A., Corliss, G.F. Automatic differentiation of algorithms: theory, implementation, and application. SIAM: Philadelphia, 1991.

- Hampton, J., and Fournier, D.A. 2001. A spatially disaggregated, length-based, age-structured population model of yellowfin tuna (*Thunnus albacares*) in the western and central Pacific Ocean. *Mar. Freshw. Res.* 52: 937–963. <https://doi.org/10.1071/MF01049>.
- Fonteneau, A. Interactions between tuna fisheries: a global review with specific examples from the Atlantic Ocean. In Status of Interactions of Pacific Tuna Fisheries in 1995. Proceedings of the Second FAO Expert Consultation on Interactions of Pacific Tuna Fisheries, Shimizu, Japan, 23–31 January 1995; R.S. Shomura, J. Majkowski, and R.F. Harman, Eds.; FAO Fisheries Technical Paper, 1996; No. 365.
- Matear, R. J. 1995. Parameter optimization and analysis of ecosystem models using simulated annealing: a case study at Station P. *Journal of Marine Research.* 1995, 53, 571–607.
- Otter Research Ltd. Autodif: a C++ array extension with automatic differentiation for use in nonlinear modeling and statistics. Otter Research Ltd: Nanaimo, Canada, 1994.
- Pacific Community (SPC). 2016. Tuna Fisheries Yearbook. Western and Central Pacific Fisheries Commission, Pohnpei, Federated States of Micronesia.
- Pianosi, F., Beven, K., Freer, J., Hall, J., Rougier, J., Stephenson, D., and Wagener, T. Sensitivity analysis of environmental models: A systematic review with practical workflow. *Environ. Modell. Softw.* 79, 214–232. <https://doi.org/10.1016/j.envsoft.2016.02.008>.
- Robinson, A.R., Lermusiaux, P. F. J. 2002. Data assimilation for modeling and predicting coupled physical biological interactions in the sea. From *The Sea*, Volume 12, edited by Allan R. Robinson, James J. McCarthy, and Brian J. Rothschild. John Wiley & Sons, Inc., New York. 475–536.
- Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., et al. 2008. Global sensitivity analysis. The Primer. John Wiley and Sons.
- Senina, I., Sibert, J., and Lehodey, P. 2008. Parameter estimation for basin-scale ecosystem-linked population models of large pelagic predators: Application to skipjack tuna. *Prog. Oceanogr.* 78: 319–335. doi:10.1016/j.pocean.2008.06.003.
- Senina, I., Lehodey, P., Hampton, J. and J. Sibert. 2020a. Quantitative modelling of the spatial dynamics of South Pacific and Atlantic albacore tuna populations. *Deep Sea Res. II* 175, doi.org/10.1016/j.dsr2.2019.104667).
- Senina, I., Lehodey, P., Sibert, J. and J. Hampton. 2020b. Integrating tagging and fisheries data into a spatial population dynamics model to improve its predictive skills. *Can. J. Aquat. Fish. Sci.* 77, 576–593.

- Senina, I., Lehodey, P., Nicol, S., Scutt Phillips, J., Hampton, J., 2020c. SEAPODYM: revisiting bigeye reference model with conventional tagging data. WCPFC-SC16-2020.
- Sibert, J.R., Hampton, J., Fournier, D.A., Bills, P.J. 1999. An advection-diffusion-reaction model for the estimation of fish movement parameters from tagging data, with application to skipjack tuna (*Katsuwonus pelamis*). *Can. J. Fish. Aquat. Sci.* 56, 925-938.
- Taylor, K.E. 2001. Summarizing multiple aspects of model performance in a single diagram. *J. Geophys. Res.* 106: 7183–7192. <https://doi.org/10.1029/2000JD900719>.
- Vallino, J.J. 2000. Improving marine ecosystem models: use of data assimilation and mesocosm experiments. *Journal of Marine Research.* 58, 117-164.
- Worley, B. 1991. Experience with the forward and reverse mode of GRESS in contaminant transport modeling and other applications. In Griewank, A. and G.F. Corliss. Automatic differentiation of algorithms: theory, practice and application. SIAM, Philadelphia.

Appendix A

Inventory of model variables and input parameters

A.1 Selected notations

Table A.1: Model variables, notations and units as used in the model code. Note, all notations in this table are given for the model configuration with monthly temporal resolution. The **units in bold** mean that the variable has a different unit as an input variable. In this case see text in the **Definition** column. Abbreviations used: m - meter, nmi - nautical mile, d - day, mo - month, Nb - number of individuals, mt - metric tonne.

Symbol	Definition	Units
<i>Spatial domain</i>		
$\Omega, \partial\Omega$	two-dimensional model domain and its complex boundary defined by the land mask	
x, y	spatial coordinates in two-dimensional model domain	digital ($^{\circ}$)
z	vertical coordinates, giving depth of three pelagic layers: (1) $0 - 1.5 \times Z_{eu}$, (2) $1.5 \times Z_{eu} - 4.5 \times Z_{eu}$ and (3) $4.5 \times Z_{eu} - \min(10 \times Z_{eu}, 1000m)$	m
<i>Environmental data</i>		
$\mathbf{v}_z(t, x, y)$	vector (u, v) of horizontal currents, averaged over vertical layer z (OGCM modelled data), it has units of $\text{m}\cdot\text{s}^{-1}$ in the input file	$\text{nmi}\cdot\text{mo}^{-1}$
$T_z(t, x, y)$	water temperature (OGCM modelled data), averaged over layer z	$^{\circ}\text{C}$
$O_z(t, x, y)$	concentration of dissolved oxygen, averaged over vertical layer z (BGCH model or Levitus database)	$\text{ml}\cdot\text{l}^{-1}$

Table A.1 continued

Symbol	Definition	Units
$P(t, x, y)$	vertically integrated primary production (either output of BGCH model or VGPM model based on satellite-derived Chl-a data), input data have units $\text{mmol C} \cdot \text{m}^{-2} \cdot \text{d}^{-1}$	$\frac{\text{mmol C}}{\text{m}^2 \cdot \text{mo}}$
<i>Coupled ADR model variables</i>		
$F_{z_d, z_n}(t, x, y)$	density of functional groupw of micronekton (food for tunas) living in pelagic layer z_d during daytime and migrating to layer z_n during night time	$\text{g} \cdot \text{m}^{-2}$
$N(a, t, x, y)$	density of tuna population at age a , time t and spatial position (x, y)	$\text{Nb} \cdot \text{km}^{-2}$
<i>Environmental (habitat) indices</i>		
$\Theta(a, x, y, z)$	accessibility of tuna of age a to the vertical layer z	
$H_s(t, x, y)$	spawning or larvae's habitat index	
$H_j(t, x, y)$	juvenile's habitat index	
$H_a(a, t, x, y)$	adult's (feeding and movement) habitat index	
<i>Advection-diffusion-reaction parameters</i>		
$\mathbf{v}(a, t, x, y)$	vector field of total tuna density velocity	$\text{nmi} \cdot \text{mo}^{-1}$
$\mathbf{v}_c(a, t, x, y)$	vector field of ocean currents computed as weighted average through all pelagic layers with weights being accessibility to the layers	$\text{nmi} \cdot \text{mo}^{-1}$
$\mathbf{v}_N(a, t, x, y)$	vector field of tuna density active velocity towards a gradient of stimuli	$\text{nmi} \cdot \text{mo}^{-1}$
$D(a, t, x, y)$	diffusion coefficient, measuring the rate of density dispersal due to random movements	$\text{nmi}^2 \cdot \text{mo}^{-1}$
$M(a, t, x, y)$	total mortality due to fishing m_F and natural causes m_N	mo^{-1}
<i>Optimization variables</i>		
C_f	total catch by fishery f	mt
$Q_{f,r}$	proportion of length frequencies for fishery f and region r	
R_k	density of tagged individuals in the population, which belong to k-th cohort	$\text{Nb} \cdot \text{km}^{-2}$
L_C^-	catch data contribution to a negative log-likelihood	
L_Q^-	LF data contribution to a negative log-likelihood	
L_R^-	tag recaptures contribution to a negative log-likelihood	
L^-	total negative log-likelihood function	
\mathbf{H}	Hessian matrix	

A.2 XML parfile

An example of SEAPODYM XML parameter file for bigeye reference model configuration with only five fisheries and tagging data with 14 cohorts of tagged and recaptured tunas.

```

<?xml version="1.0"?>
<par version="1.0">

<!--=====
<!--SIMULATION PARAMETERS-->
<!--===== -->
<latitudeMin value="-54.875"/>
<latitudeMax value="65.125"/>
<longitudeMin value="88.375"/>
<longitudeMax value="290.375"/>
<space_reso value="2"/>
<nb_layer value="3"/>
<deltaX value="120"/>
<deltaY value="120"/>
<deltaT value="30"/>
<iterationNumber value="30"/>
<tuna_spinup value="0"/>
<save_first_date year="1998" month="1"/>
<save_last_date year="2009" month="12"/>
<nb_yr_forecast value="0"/>
<nb_step_to_skip value="36"/>
<!--=====
<!--LIKELIHOOD PARAMETERS-->
<!--===== -->
<tag_likelihood_only value="0"/>

<!--type of data in the likelihood: 0 - cpue, 1 - catch -->
<like_c_cpue value="1"/>

<total_likelihood value="1375175.761915711"/>

<!--likelihood types: 1 - concentrated , 2 - log-normal , 3 - poisson , 4 -
     negative binomial , 5 - zero-inflated negbin , 6 - truncated poisson -->
<!--can be different for each fishery (see below the fishery codes) -->
<likelihood_types>
    <bet>3 3 3 1</bet>
</likelihood_types>

<nb_fishery value="5"/>
<list_fishery_name>L1 L2 L3 L4 L5</list_fishery_name>

<!--1 - C in metric tones; 0 - C in numbers -->

```

```

<fishery_catch_units>1 1 1 1 1</fishery_catch_units>

<degrade_fishery_reso_deg value="5"/>

<!--Mask for fishery data usage: 0 – do not take into account-->
<mask_fishery_sp>
  <bet>1 1 1 1 1</bet>
</mask_fishery_sp>

<mask_fishery_no_effort>
  <bet>0 0 0 0 1</bet>
</mask_fishery_no_effort>

<mask_fishery_likelihood>
  <bet>1 1 1 1 1</bet>
</mask_fishery_likelihood>

<!--length frequencies likelihood: 0 – do not take into account , 1 – account
     -->
<frq_likelihood bet="0"/>

<!--TAG likelihood: 0 – do not take into account , 1 – account-->
<tag_likelihood bet="0"/>

<!--STOCK likelihood: 0 – do not take into account , 1 – account-->
<stock_likelihood bet="1"/>
<!-- The values entered for mean_stock_obs will be used in stock likelihood
      These can be the total abundance estimated by stock assessment models
      within the rectangular region specified -->
<mean_stock_obs>
  <bet value="1500" lgmin="110" lgmax="210" ltmin="-40" ltmax="50"/>
</mean_stock_obs>

<!--===== -->
<!--MODEL INPUT-OUTPUT-->
<!--===== -->

<!-- Forcing data directory -->
<strdir value="/data/run-freeglorys-2x30d-1998-2019-po/" />

<!-- Micronekton density files within forcing directory -->
<strdir_forage value="forage/" />
<!-- Model initial state directory containing bet_cohorts.dym -->
<strdir_init value="init/bet/19971215/e27.1_clt/e27.1.M/F0/" />

<!-- Land mask and topographic index files -->
<str_file_mask value="mask_freeglorys_2deg_corrected_noIO.txt" />
<str_file_topo value="topo_noeff.txt" />

```

```

<!--type of oxygen data: 0 for time series , 1 - for monthly , 2 for seasonal-->
<type_oxy value="1"/>

<!-- Forcing data filenames -->
<strfile_pp value="po_freeglorys2v4_rxd_2x30d_pp_1998_2019.dym"/>
<strfile_sst value="po_freeglorys2v4_2x30d_SST_1998_2019.dym"/>
<strfile_vld value="po_freeglorys2v4_rxd_2x30d_zeu_1998_2019.dym"/>
<strfile_u layer0="po_freeglorys2v4_rxd_2x30d_U_L1_1998_2019.dym" layer1=
    po_freeglorys2v4_rxd_2x30d_U_L2_1998_2019.dym" layer2=
    po_freeglorys2v4_rxd_2x30d_U_L3_1998_2019.dym"/>
<strfile_v layer0="po_freeglorys2v4_rxd_2x30d_V_L1_1998_2019.dym" layer1=
    po_freeglorys2v4_rxd_2x30d_V_L2_1998_2019.dym" layer2=
    po_freeglorys2v4_rxd_2x30d_V_L3_1998_2019.dym"/>
<strfile_t layer0="po_freeglorys2v4_rxd_2x30d_temperature_L1_1998_2019.dym"
    layer1="po_freeglorys2v4_rxd_2x30d_temperature_L2_1998_2019.dym" layer2=
    po_freeglorys2v4_rxd_2x30d_temperature_L3_1998_2019.dym"/>
<strfile_oxy layer0="po_levitus_WOA2013_2x30d_O2_L1_CLM.dym" layer1=
    po_levitus_WOA2013_2x30d_O2_L2_CLM.dym" layer2=
    po_levitus_WOA2013_2x30d_O2_L3_CLM.dym"/>

<strfile_pp_clm value="po_freeglorys2v4_rxd_2x30d_PP_1980.dym"/>
<strfile_u_clm layer0="po_freeglorys2v4_rxd_2x30d_U_L1_1980.dym" layer1=
    po_freeglorys2v4_rxd_2x30d_U_L2_1980.dym" layer2=
    po_freeglorys2v4_rxd_2x30d_U_L3_1980.dym"/>
<strfile_v_clm layer0="po_freeglorys2v4_rxd_2x30d_V_L1_1980.dym" layer1=
    po_freeglorys2v4_rxd_2x30d_V_L2_1980.dym" layer2=
    po_freeglorys2v4_rxd_2x30d_V_L3_1980.dym"/>
<strfile_t_clm layer0="po_freeglorys2v4_rxd_2x30d_temperature_L1_1980.dym"
    layer1="po_freeglorys2v4_rxd_2x30d_temperature_L2_1980.dym" layer2=
    po_freeglorys2v4_rxd_2x30d_temperature_L3_1980.dym"/>
<strfile_oxy_clm layer0="po_freeglorys2v4_rxd_2x30d_O2_L1_1980.dym" layer1=
    po_freeglorys2v4_rxd_2x30d_O2_L2_1980.dym" layer2=
    po_freeglorys2v4_rxd_2x30d_O2_L3_1980.dym"/>

<!--FISHERY DATA -->
<strdir_fisheries value="/data/fisheries/bet/ds2020 /"/>

<file_catch_data>
    <nb_files bet="1"/>
        <bet file1="bet_catch_20f_L3cor.txt"/>
</file_catch_data>

<flex_Regstruc value="1"/>
<file_frq_data>
    <nb_files bet="2"/>
        <bet file1="bet_LF_LL_11f_PO.txt"/>
        <bet file2="bet_LF_PS_PL_OT_9f_PO.txt"/>
</file_frq_data>

```

```

<!--TAGGING DATA -->
<strdir_tags value="/data/tags/ds2016/" />

<tags_grid>
    <reso dx="6" dy="6"/>
    <longitude east="110.375" west="290.375"/>
    <latitude south="-30.625" north="29.375"/>
</tags_grid>

<file_tag_data>
    <nb_files bet="14"/>
    <bet file1="bet-pop-tags_20080715.txt" file2="bet-pop-tags_20080815.txt"
        file3="bet-pop-tags_20080915.txt" file4="bet-pop-tags_20081015.txt"
        file5="bet-pop-tags_20081115.txt" file6="bet-pop-tags_20081215.txt"
        file7="bet-pop-tags_20090515.txt" file8="bet-pop-tags_20090615.txt"
        file9="bet-pop-tags_20090715.txt" file10="bet-pop-tags_20090815.txt"
        file11="bet-pop-tags_20090915.txt" file12="bet-pop-tags_20091015.txt"
        file13="bet-pop-tags_20091115.txt" file14="bet-pop-tags_20091215.txt"/>
</file_tag_data>

<!-- Output Directory and writing options -->
<strdir_output value="output/" />
<write_all_fisheries_dym value="1" />
<write_all_cohorts_dym value="1" />

<!--FORAGE MODEL PARAMETERS-->
<!--FORAGE MODEL PARAMETERS-->
<!--FORAGE MODEL PARAMETERS-->

<Tr_max value="527" />
<Tr_exp value="-0.125" />
<inv_lambda_max value="2109" />
<inv_lambda_curv value="-0.125" />
<E value="0.0036972" />
<c_pp value="0.0948" />
<nb_forage value="6" />

<frg_name>epi meso mmeso bathy mbathy hmbathy</frg_name>

<source_frg epi="0.08" meso="0.17" mmeso="0.08" bathy="0.56" mbathy="0.07"
    hmbathy="0.04" />
<day_layer epi="0" meso="1" mmeso="1" bathy="2" mbathy="2" hmbathy="2" />
<night_layer epi="0" meso="1" mmeso="0" bathy="2" mbathy="1" hmbathy="0" />

<tstep_forage value="30" />
<sigma_fcte value="100" />

```

```

<max_NPP value="200"/>
<b_F_habitat value="20000"/>

<!--—————→
<!--PREDATOR MODEL AGE STRUCTURE-->
<!--—————→

<nb_species value="1"/>

<sp_name>bet</sp_name>

<nb_life_stages bet="3"/>
<life_stage>
    <bet>larvae juvenile adult</bet>
</life_stage>

<nb_cohort_life_stage>
    <bet>1 2 82</bet>
</nb_cohort_life_stage>

<age_recruit bet="3"/>
<age_mature bet="35"/>

<sp_unit_cohort>
    <bet>30   30   30   30   30   30   30   30   30   30   30   30   30   30   30   30
        30   30   30   30   30   30   30   30   30   30   30   30   30   30   30   30
        30   30   30   30   30   30   30   30   30   30   30   30   30   30   30   30
        30   30   30   30   30   30   30   30   30   30   30   30   30   30   30   30
        30   30   30   30   30   30   30   30   30   30   30   30   30   30   30   30
        30   30   30   30   30   30   30   30   30   30   2520
    </bet>
</sp_unit_cohort>

<!--Just for info , the mean age (in month) of each age class -->
<!--age = 1   3   5   7   9   11  13   15   17   19   21   23   25   27   29   31   33
    35   37
        39   41   43   45   47   49   51   53   55   57   59   61   63   65   67   69   71
        73   75
        77   79   81   83   85   87   89   91   93   95   138
-->
<maturity_age>
    <bet>0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
        0.0001  0.0002  0.0004  0.0007  0.0011  0.0018  0.0030  0.0042  0.0066  0.0101
        0.0136  0.0199  0.0289  0.0378  0.0519  0.0712  0.0905  0.1169  0.1505  0.1841
        0.2240  0.2702  0.3164  0.3647  0.4149  0.4651  0.5128  0.5579  0.6030  0.6434
        0.6792  0.7149  0.7461  0.7727  0.7993  0.8223  0.8415  0.8608  0.8774  0.8912
        0.9051  0.9170  0.9270  0.9370  0.9456  0.9527  0.9599  0.9661  0.9712  0.9763
        0.9806  0.9840  0.9875  0.9904  0.9926  0.9948  0.9964  0.9976  0.9987  0.9994
        0.9996  0.9999  0.9997  0.9992  0.9987  0.9978  0.9966  0.9954  0.9939  0.9921
        0.9903  0.9882  0.9859  0.9836  0.7300
    </bet>

```

```

</bet>
</maturity_age>

<!--Optional: if active , vector age_compute_habitat gives the sequence number
of
updates of feeding habitat and movement rates. For instance , in this
parfile
these internal variables will be computed (updated) only 32 times instead
of
85, i.e. the number of age classes-->
<!--
<age_compute_habitat>
<bet>0 1 2 3 4 5 5 6 6 7 7 8 8 9 9 10 10 11 11 12 12 13 13 14
14
15 15 16 16 17 17 18 18 19 19 19 20 20 20 21 21 21 22 22 22 23 23 24
24
24 24 25 25 25 25 26 26 26 27 27 27 27 27 28 28 28 28 28 29 29 29
29
29 30 30 30 30 30 30 30 31 32
</bet>
</age_compute_habitat>
__>
<!--mean length (cm) of individuals in each age class-->
<length>
<bet>9.12 13.55 17.88 22.10 26.21 30.22 34.14 37.95 41.68 45.31
48.85 52.30 55.67 58.95 62.16 65.28 68.33 71.30 74.20 77.03
79.79 82.48 85.10 87.66 90.16 92.59 94.96 97.28 99.54 101.74
103.89 105.98 108.02 110.02 111.96 113.85 115.70 117.51 119.26 120.98
122.65 124.28 125.88 127.43 128.94 130.42 131.86 133.26 134.63 135.97
137.27 138.54 139.78 140.99 142.17 143.31 144.44 145.53 146.60 147.64
148.65 149.64 150.61 151.55 152.46 153.36 154.23 155.08 155.92 156.73
157.52 158.29 159.04 159.77 160.49 161.18 161.86 162.53 163.17 163.80
164.42 165.02 165.61 166.18 180.93
</bet>
</length>
<!--mean weight (kg) of individuals in each age class-->
<weight>
<bet>0.02 0.05 0.12 0.23 0.39 0.59 0.86 1.18 1.57 2.02
2.53 3.11 3.76 4.47 5.25 6.09 6.99 7.95 8.97 10.04
11.17 12.34 13.57 14.84 16.16 17.51 18.91 20.34 21.80 23.29
24.81 26.35 27.92 29.51 31.11 32.73 34.36 36.01 37.66 39.33
40.99 42.67 44.34 46.01 47.69 49.36 51.02 52.69 54.34 55.99
57.63 59.25 60.87 62.48 64.07 65.65 67.22 68.77 70.30 71.82
73.33 74.81 76.28 77.73 79.17 80.58 81.98 83.35 84.71 86.05
87.37 88.67 89.95 91.21 92.45 93.67 94.87 96.05 97.21 98.35
99.47 100.57 101.66 102.72 132.87
</bet>
</weight>

```

```
<!--—————>
<!--PREDATOR MODEL PARAMETERS-->
<!--—————>

<!--== 1. Available OPTIONS ==-->
<seasonal_migrations bet="0"/>

<vertical_movement bet="1"/>
<!-- if optimal mu and sigma (b_sst_larvae and a_sst_larvae)
     for larvae assumed different from b_ and a_sst_spawning -->
<uncouple_sst_larvae bet="1"/>

<gaussian_thermal_function bet="1"/>

<cannibalism_juv bet="0"/>

<food_requirement_in_mortality bet="0"/>

<!-- end of Available OPTIONS ==-->

<!--== 2. Spawning habitat parameters ==-->
<a_sst_spawning bet="5.499999377710821">
    <variable min="0.5" max="5.5" use="false"/>
</a_sst_spawning>

<b_sst_spawning bet="25.50000000432488">
    <variable min="25.5" max="31.0" use="false"/>
</b_sst_spawning>

<!--parameters *_sst_larvae are ignored if
     <uncouple_sst_larvae bet="0"/> -->
<a_sst_larvae bet="3.749999993906406">
    <variable min="0.5" max="3.75" use="false"/>
</a_sst_larvae>

<b_sst_larvae bet="26.00001467106519">
    <variable min="26.0" max="28.0" use="false"/>
</b_sst_larvae>

<alpha_hsp_prey bet="0.02421327707107811">
    <variable min="0.002" max="0.1" use="false"/>
</alpha_hsp_prey>

<alpha_hsp_predator bet="0.5000024599599174">
    <variable min="0.5" max="1.5" use="false"/>
</alpha_hsp_predator>

<beta_hsp_predator bet="0.7696338480923586">
```

```

<variable min="0.0" max="2.0" use="false" />
</beta_hsp_predator>
<!-- end of Spawning habitat parameters -->

<!-- 3. Recruitment function -->
<nb_recruitment bet="0.006500965085506141">
    <variable min="0.0005" max="0.05" use="true" />
</nb_recruitment>

<a_adults_spawning bet="15.72612509571025">
    <variable min="10" max="20" use="true" />
</a_adults_spawning>
<!-- end of Recruitment function -->

<!-- 4. Juvenile habitat parameters -->
<!-- Activated by flag <cannibalism_juv bet="1"/> -->
<hp_cannibalism bet="5.0">
    <variable min="0" max="20" use="false" />
</hp_cannibalism>
<!-- end of Juvenile habitat parameters -->

<!-- 5. Adult seasonal migrations -->
<!-- Activated if <seasonal_migrations bet="1"/> -->
<spawning_season_peak bet="180.86071670988996">
    <variable min="0" max="100" use="false" />
</spawning_season_peak>

<spawning_season_start bet="1.00477324755014">
    <variable min="0.95" max="1.2" use="false" />
</spawning_season_start>
<!-- end of Adult seasonal migrations -->

<!-- 6. Adult feeding habitat -->
<a_sst_habitat bet="2.434728444404373">
    <variable min="0.5" max="7.5" use="false" />
</a_sst_habitat>

<b_sst_habitat bet="15.48493201609903">
    <variable min="9.0" max="15.5" use="false" />
</b_sst_habitat>

<T_age_size_slope bet="0.7000001820865549">
    <variable min="0.7" max="3.1" use="false" />
</T_age_size_slope>

<!-- three parameters below are active only if
    <gaussian_thermal_function bet="0"/> -->
<thermal_func>
    <delta1 bet="0.04" />

```

```

        <variable min="0.0" max="0.05" use="false" />
    </delta1>

    <delta2 bet="0.015">
        <variable min="0.0" max="0.25" use="false" />
    </delta2>

    <delta3 bet="0.25">
        <variable min="0.0" max="10.0" use="false" />
    </delta3>
</thermal_func>

<a_oxy_habitat bet="0.0001120486069333051">
    <variable min="0.00001" max="0.15" use="false" />
</a_oxy_habitat>

<b_oxy_habitat bet="1.485492424315652">
    <variable min="0.5" max="2.5" use="false" />
</b_oxy_habitat>

<eF_habitat>
    <epi bet="0.99999999843883">
        <variable min="0.0" max="1.0" use="false" />
    </epi>

    <meso bet="0.4858662333732704">
        <variable min="0.0" max="1.5" use="false" />
    </meso>

    <mmeso bet="0.999997359387802">
        <variable min="0.0" max="1.0" use="false" />
    </mmeso>

    <bathy bet="0.0">
        <variable min="0.0" max="1.0" use="false" />
    </bathy>

    <mbathy bet="0.3653131477844179">
        <variable min="0.0" max="1.0" use="false" />
    </mbathy>

    <hmbathy bet="0.999999997564847">
        <variable min="0.0" max="1.0" use="false" />
    </hmbathy>
</eF_habitat>
<!-- end of Adult feeding habitat -->

<!-- 7. Movement parameters -->
<sigma_species bet="1.054119672277597">
```

```

<variable min="0" max="1.5" use="false" />
</sigma_species>

<c_diff_fish bet="0.9">
  <variable min="0.0" max="1.0" use="false" />
</c_diff_fish>

<MSS_species bet="0.150000000076028">
  <variable min="0.15" max="2.0" use="false" />
</MSS_species>

<MSS_size_slope bet="0.7500119452996615">
  <variable min="0.5" max="1.1" use="false" />
  </MSS_size_slope>
<!-- end of Movement parameters -->

<!-- 8. Mortality parameters -->
<Mp_mean_max bet="0.1135833270984567">
  <variable min="0.0" max="0.12" use="false" />
  </Mp_mean_max>

<Mp_mean_exp bet="0.02306">
  <variable min="0.01" max="0.5" use="false" />
  </Mp_mean_exp>

<Ms_mean_max bet="1.0e-10">
  <variable min="0.0" max="0.05" use="false" />
  </Ms_mean_max>

<Ms_mean_slope bet="3.85">
  <variable min="0.0" max="5.05" use="false" />
  </Ms_mean_slope>

<M_mean_range bet="0.1">
  <variable min="0.0" max="0.5" use="false" />
  </M_mean_range>
<!-- end of Mortality parameters -->

<!-- 9. Food requirement index -->
<!-- Used only if <food_requirement_in_mortality bet="1"/>-->
<!-- residual competition is the proportion of total F biomass
available for a given species in the presence of other species -->
<residual_competition bet="0.025" />

<!-- daily food ration -->
<forage_ration bet="0.06" />
<!-- end of Food requirement index -->

```

```

<!--===== -->
<!--FISHERY DATA AND PARAMETERS-->
<!--===== -->

<q_sp_fishery>
<variables use="true"/>

    <L1 bet="1.298725739770595e-05">
        <variable min="0.0" max="0.001" use="true" dyn="0.004"/>
    </L1>

    <L2 bet="2.57024034053101e-05">
        <variable min="0.0" max="0.001" use="true" dyn="0.004"/>
    </L2>

    <L3 bet="8.240793041768003e-06">
        <variable min="0.0" max="0.001" use="true" dyn="0.004"/>
    </L3>

    <L4 bet="1.73244042928708e-05">
        <variable min="0.0" max="0.001" use="true" dyn="5e-04"/>
    </L4>

    <L5 bet="0.0">
        <variable min="0.0" max="0.001" use="false" dyn="0.0"/>
    </L5>
</q_sp_fishery>

<!--selectivities as functions of length-->
<!--1 - logistic , 1 parameter: main parameter is used as a slope coefficient
     -->
<!--2 - sigmoid , 2 parameters: main parameter as slope coefficient +
      length_threshold-->
<!--3 - asymmetric Gaussian , 3 parameters: length_threshold as optimal value ,
      main as sigma , right_asymptote as the function min on the right -->
<s_sp_fishery>
<variables use="true"/>

    <L1 bet="23.79737028104424">
        <variable min="5" max="50" use="true"/>
        <function_type value="3"/>
        <length_threshold bet="114.8134593861857" use="true"/>
        <right_asymptote bet="0.9999999999999976" min="0" max="1" use
                         ="false"/>
    </L1>
    <L2 bet="19.5874705020817">
        <variable min="5" max="60" use="true"/>
        <function_type value="3"/>
        <length_threshold bet="126.6702726553326" use="true"/>

```

```

        <right_asymptote bet="0.25" min="0.25" max="1" use="false" />
    </L2>
    <L3 bet="0.13">
        <variable min="0.05" max="0.6" use="false" />
        <function_type value="2" />
        <length_threshold bet="102.0580630708797" use="true" />
        <right_asymptote bet="0.4" min="0" max="1" use="false" />
    </L3>
    <L4 bet="18.86567062907572">
        <variable min="5" max="36" use="true" />
        <function_type value="3" />
        <length_threshold bet="130.7283846873609" use="true" />
        <right_asymptote bet="0.2294335924829242" min="0" max="1" use="true" />
    </L4>
    <L5 bet="0.1694198322792639">
        <variable min="0.05" max="0.6" use="true" />
        <function_type value="2" />
        <length_threshold bet="70.86730272234507" use="true" />
        <right_asymptote bet="0.7" min="0" max="1" use="false" />
    </L5>
</s_sp_fishery>

<!--===== -->
<!--AGGREGATION ZONES-->
<!--===== -->
<nb_region value="11"/>
<use_mask_catch value="0" />
<area0 area_id="1" lgmin="120" lgmax="170" ltmin="20" ltmax="50" />
<area1 area_id="2" lgmin="170" lgmax="210" ltmin="10" ltmax="50" />
<area2 area_id="3" lgmin="140" lgmax="170" ltmin="-10" ltmax="10" />
<area3 area_id="4" lgmin="170" lgmax="210" ltmin="-10" ltmax="10" />
<area4 area_id="5" lgmin="140" lgmax="170" ltmin="-40" ltmax="-10" />
<area5 area_id="6" lgmin="170" lgmax="210" ltmin="-40" ltmax="-10" />
<area6 area_id="7" lgmin="110" lgmax="140" ltmin="-10" ltmax="20" />
<area7 area_id="8" lgmin="210" lgmax="240" ltmin="-40" ltmax="10" />
<area8 area_id="9" lgmin="240" lgmax="290" ltmin="-40" ltmax="10" />
<area9 area_id="10" lgmin="210" lgmax="280" ltmin="10" ltmax="50" />
<area10 area_id="11" lgmin="140" lgmax="170" ltmin="10" ltmax="20" />

<nb_region_sp_B bet="11" />
<area_sp_B>
    <bet>1 2 3 4 5 6 7 8 9 10 11</bet>
</area_sp_B>

<nb_EEZ value="0" />
<!--File maskEEZ should be placed in input data directory-->
<str_file_maskEEZ value="SP_IPSL_mask_eez_wcpfc.txt" />
<EEZ>
```

```
<eez0 name="New Caledonia" id="-19"/>
<eez1 name="Fiji" id="-12"/>
<eez2 name="Vanuatu" id="-33"/>
<eez3 name="Solomon Islands" id="-29"/>
<eez4 name="New Zealand" id="-23"/>
<eez5 name="Norfolk" id="-20"/>
<eez6 name="Matthew Hunter" id="-16"/>
</EEZ>

<!-- Parameters likelihood_parameters and prob_zero are
     parameters of zero-inflated negative binomial likelihood -->
<likelihood_parameters>
<variables use="false"/>
<L1 bet="0.01"/>
  <L2 bet="0.007"/>
  <L3 bet="0.0"/>
  <L4 bet="0.01"/>
<L5 bet="0.01"/>
</likelihood_parameters>

<prob_zero>
<L1 bet="0.0"/>
  <L2 bet="0.0"/>
  <L3 bet="0.0"/>
  <L4 bet="0.0"/>
<L5 bet="0.0"/>
</prob_zero>

</par>
```


Appendix B

Technical annex

B.1 Makefile

```
HEADERS= \
XMLDocument2.h \
calpop.h \
Map.h \
Matrices.h \
Numfunc.h \
ReadWrite.h \
Param.h \
Date.h \
SaveTimeArea.h \
SeapodymCoupled.h \
SimtunaFunc.h \
mytypes.h \
SeapodymDocConsole.h \
ctrace.h \
Utilities.h \
VarMatrices.h \
VarParamCoupled.h \
VarSimtunaFunc.h

SRCS= \
main.cpp \
XMLDocument2.cpp \
VarParamCoupled.cpp \
seapodym_coupled.cpp \
```

```
Map.cpp \
Matrices.cpp \
Numfunc.cpp \
ReadWrite.cpp \
Param.cpp \
Date.cpp \
SaveTimeArea.cpp \
SimtunaFunc.cpp \
VarParamCoupled_xinit.cpp \
VarParamCoupled_reset.cpp \
SeapodymCoupled_EditRunCoupled.cpp \
SeapodymCoupled_OnRunCoupled.cpp \
SeapodymCoupled_OnRunFirstStep.cpp \
SeapodymCoupled_OnReadForcing.cpp \
SeapodymCoupled_OnWriteOutput.cpp \
SeapodymDocConsole_UpdateDisplay.cpp \
dv_spawning_habitat.cpp \
dv_juvenile_habitat.cpp \
dv_mortality_sp.cpp \
dv_spawning_habitat.cpp \
dv_caldia.cpp \
dv_tridag_bet.cpp \
dv_calrec_adre.cpp \
dv_survival.cpp \
dv_precalrec_juv.cpp \
dv_calrec_precalrec.cpp \
dv_predicted_catch.cpp \
dv_predicted_catch_without_effort.cpp \
dv_total_exploited_biomass.cpp \
dv_total_obs_catch_age.cpp \
dv_spawning.cpp \
dv_accessibility.cpp \
dv_feeding_habitat.cpp \
dv_seadonal_switch.cpp \
dv_total_pop.cpp \
dv_total_mortality_comp.cpp \
dv_food_requirement_index.cpp \
Calpop_caldia.cpp \
Calpop_calrec.cpp \
Calpop_InitCalPop.cpp \
Calpop_precaldia.cpp \
```

```
Calpop_precalrec.cpp \
Calpop_tridag.cpp \
Calpop_recompute_coefs.cpp \
like.cpp

SRCPATH=DOM/src:src
INCPATH=-IDOM/src -Isrc
BINPATH=.

OBJPATH=objs
ADMODEL=$(ADMB_HOME)

DEBUG= -g
CFLAGS=-DTRUE=true -DFALSE=false -D __GNUDOS__ -Dlinux -O3 -DOPT_LIB
-Wall -Wno-deprecated -I$(ADMODEL)/include -I/usr/include/libxml2 $(INCPATH)
LFLAGS= -L$(ADMODEL)/lib -ladmbo -ldl -lstdc++ -lxml2 -lm
CC=gcc
LL=$(CC)

vpath %.cpp $(SRCPATH)
vpath %.h $(SRCPATH)

OBJECTS=$(SRCS:.cpp=$(OBJPATH)/%.o)

export OBJECTS
export OBJPATH

export CFLAGS
export LFLAGS
export CC
export LL

all: init $(BINPATH)/seapodym

init:
    @test -d $(OBJPATH) || mkdir -v $(OBJPATH)

test: init $(OBJECTS)
    make -f Makefile.test

docs: $(SRCS) $(HEADERS)
```

@doxygen

```
$(BINPATH)/seapodym : $(OBJECTS)
$(LL) -o$@ $(DEBUG) $^ $(LFLAGS)

$(OBJPATH)/%.o : %.cpp
$(CC) -o$@ $(DEBUG) -c $(CFLAGS) $(filter %.cpp, $^)

$(OBJECTS) : $(HEADERS)
```

clean:

```
@rm -vf $(OBJECTS) $(BINPATH)/seapodym
@rm -vf $(OBJECTS) $(BINPATH)/seapodym.exe
@rm -vf $(OBJECTS) $(BINPATH)/TestSeapodymCoupled
@rm -vrf $(OBJPATH)
@rm -vrf docs
@rm -vf gmon.out
```

Appendix C

File formats

C.1 ASCII fishing data files

C.1.1 Effort and catch

The raw geo-referenced effort and catch data is a voluminous dataset containing the information on fishery, gear, date, lon-lat coordinates, spatial resolution as well as corresponding records of effort and catch. In order to be correctly interpreted by the SEAPODYM software, the data should be organised and formatted as described below. Note that it is advised to keep the original spatial resolution of fisheries data. This avoids unnecessary inflation of the ASCII file and allows fitting the catch at original spatial resolution. As described in Chapter 4 (section 4.1.1), the uniform distribution of fishing effort or catch is implemented in the code in order to ensure that the fishing mortality is always applied at the spatial resolution of the model.

The effort and catch data file format is shown below. It is a tabulation-separated ASCII file with extension .txt, having a small header followed by the table with row names as follows (example):

7	← number of fisheries
84286 52530 47545 21675 3354 158510 120459	← number of records by fishery
f yr mm dd gr lat lon res E C	← column names
1 1950 8 15 L 7.5 152.5 5 60 1.118	← the first record in the file
...	

The detailed description of each record is provided in Table C.1. Note, the name of the fishery is read by the code as the concatenation of letter in **gr** and the ID in **f**. For example, the record above belongs to fishery 'L1'. Hence, each unique combination of gear and ID is interpreted as a stand-alone fishery and the number of fisheries as well as number of records

Table C.1: The information written in the input geo-referenced effort and catch data file prepared for the SEAPODYM software.

Column name	Description	Type
f	Fishery ID	integer
yr	Year of catch	integer
mm	Month of catch	integer
dd	Day of catch	integer
gr	Usually a single letter denoting fishing gear. Examples: L for long-line, S for purse-seine, P for pole and line, T for troll, O for other (combination of different gears)	character
lat	Latitudinal coordinate of the centre of the square, in which the catch data was aggregated, given in digital degrees	float
lon	Longitudinal coordinate of the centre of the square, in which the catch data was aggregated, given in digital degrees	float
res	The size of the square in which the catch data was aggregated, in degrees; should be unique by fishery	float
E	Fishing effort, can be given in different units; the units are implicit (affect the catchability coefficients) and should be unique within a given fishery	float
C	Catch of a given species; the units, usually metric tonnes or number of individuals, are not stored in this file and should be unique through all fisheries in the file	float

for each fishery must correspond to the information provided in the header. Consequently, all fisheries present in this file have to be defined in the configuration parfile in order to be taken into account.

C.1.2 Size frequencies

Although it is strongly recommended to use the length frequency (LF) data in SEAPODYM to facilitate parameter estimation, it is possible to run SEAPODYM without size frequency data by setting flag `frq_likelihood` and number of LF files `file_frq_data` to 0. Since the LF data are provided at coarse spatio-temporal resolution, usually over quarter and rectangular region, and because the data are size structured, the file format is very different

from effort and catch data file. However, the same fisheries, partially or entirely, must be present in the LF file. Besides, the region of LF record must contain at least one record with effort and catch data to be taken into account.

Several LF files can be provided as the SEAPODYM input. The splitting is necessary once the size structure of the data is not homogeneous. For example, the catch at size provided for at 2 cm bins cannot be combined with the data at 1 cm bins. Each LF file consists of the header with regional structure of the LF data and the table, which contains the information on date, region, fishery and the catch at size records. The LF file is the tabulation-separated ASCII file with extension .txt having the following structure (example):

```

668 7 20809      ← nb. of unique regions nb. of fisheries total nb. of records
1 110 115 15 20 ← region ID regional coordinates
...
11 20 5          ← number of bins first bin size bin size
yr qtr mo f reg LF[1] .. LF[11]   ← column names of the main table
1955 1 1 L1 2 0 0 0 0 5 37 19 18 4 1 0 ← standard LF record
...

```

where regional coordinates should be written as tab-separated (lon-west, lon-east, lat-south, lat-north) set. The detailed description of the LF record is provided in Table C.2. Note that since the LF data are usually quarterly, column 'month' is redundant and contains the first month of quarter, however it can potentially be used in case if higher resolution data will be available.

Table C.2: Structure of `length frequencies` data file.

Column name	Description	Type
<code>yr</code>	The year of the LF sample	integer
<code>qtr</code>	The quarter of the LF sample	integer
<code>mo</code>	The month of the LF sample or the first month of the quarter for seasonal data	integer
<code>f</code>	Here the fishery name as in the effort and catch file and in the parfile	character
<code>reg</code>	The ID of the region of a given sample	integer
<code>LF[i]</code>	The number of fish caught in the size bin i , this number can be raised to total catch	float

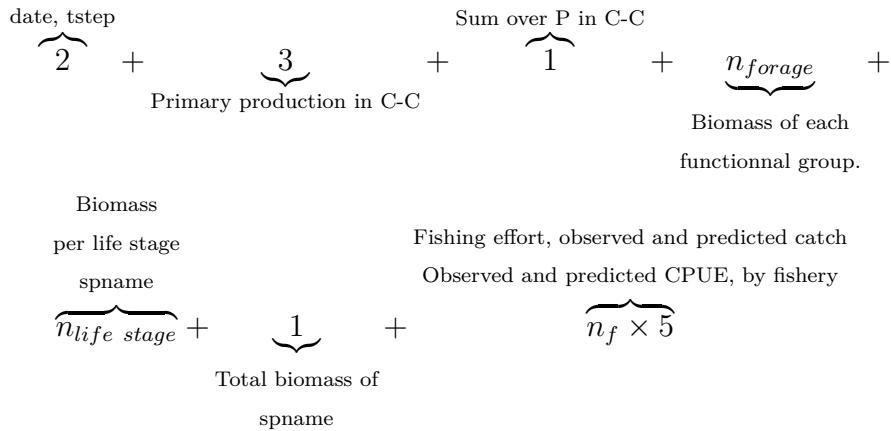


Figure C1: The structure of SumDym.txt file. The coordinates $C - C$ correspond to three regional aggregations of the primary production P : $10^{\circ}\text{N}-45^{\circ}\text{N}$, $10^{\circ}\text{S}-10^{\circ}\text{N}$ and $35^{\circ}\text{S}-10^{\circ}\text{S}$, where longitudinal extension corresponds to that of model domain. The number of columns n_{forage} , $n_{life\ stage}$, n_f denote the number of micronekton groups, the number of life stages and the number of fisheries respectively.

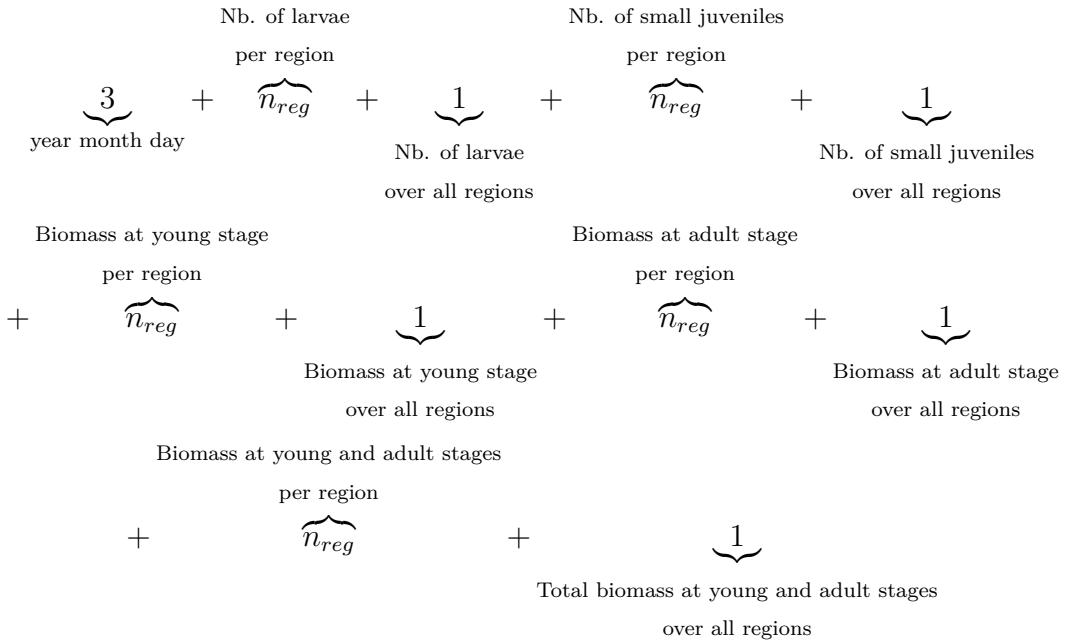
C.2 ASCII output files

C.2.1 SumDym.txt

All ASCII output files in SEAPODYM are tabulation-separated. The SumDym.txt file contains the time evolution of the model variables aggregated over the model domain, i.e. for each variable $\phi(x, y, t)$ either used as model forcing or computed by the model it has $\sum_{x,y} \phi(x, y, t)$. The scheme in Figure C1 illustrates the structure of this file. The data are written in tabular form. The one-line header contains the names of the table columns. The dimension, i.e. number of columns and rows in this file depends on the current run configuration specified in the parfile, with the number of rows being the number of time steps n_t , and number of columns depending on number of life stages and number of fisheries.

C.2.2 SumQArea.txt

This file is similar to SumDym.txt, however instead of aggregation over the model domain, it contains model variables aggregated over regional structure defined in the parfile (see section 3.4.7 for more details). Also, this file does not contain fisheries statistics. The file header provides essential details on regional structure defined in the current run and the definition of life stages. Thus, first n_{reg} lines are filled with regional coordinates corresponding to the corners of rectangular regions, written as follows: westmost longitude, eastmost

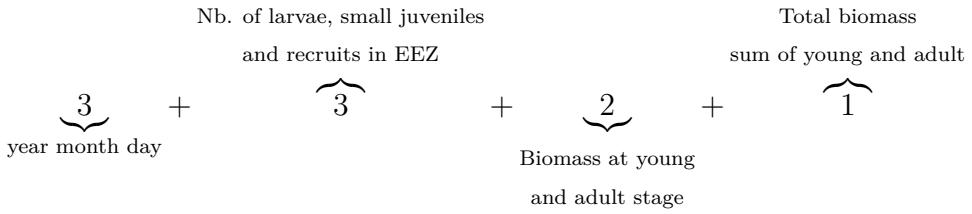
Figure C2: The structure of `SumQArea.txt` file.

longitude, southern latitude, northern latitude. Regions are followed by the summary of the species life stages among larvae, juvenile, young and adult as well as the indices of age classes included within each life stage. Note, since this file is mostly used for comparisons with regional stock assessment estimates, recruits are also written, although it is not a life stage in SEAPODYM, but simply an age class that corresponds to the definition of recruitment to the exploited population used in the stock assessment model.

The main table has the header denoting column names. The number of rows of this table corresponds to the number of model time steps, n_t , and columns as shown schematically in Figure C2.

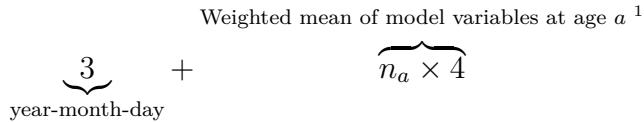
C.2.3 `SumEEZ.txt`

The tabulation-separated file `ttfamily SumEEZ_EEZname.txt` contains the time evolution of biomass extractions over selected EEZ area. The structure of this file is shown in Figure C3. The number of columns is fixed, and the aggregated statistics is computed for larvae, small juveniles, recruits (same definition as for `SumQArea.txt`) young (immature adults) and mature adults. The last column contains the sum of biomass at young and adult life stages.

Figure C3: The structure of `SumEEZ.txt` file.

C.2.4 MeanVar.txt

This ASCII output file `spname_MeanVar.txt` is used to analyse the temporal evolution of aggregated model variables at age, with the aggregation method being the weighted mean and weights being the spatial distribution of population density. The aggregated variables are: 1) natural mortality at age, 2) species speed at age, 3) species diffusion rate at age, and 4) the ambient water temperature at age. The information is structured into a matrix of n_t rows and the number of columns depending on the number of age classes. See Figure C4 for more details.

Figure C4: The structure of `spname_MeanVar.txt` file.

C.2.5 Spatial_Corr.txt

The ASCII file `spname_Spatial_Corr.txt` contains correlation statistics for catch and CPUE data. The number of rows in this file is equal to the number of time steps n_t , and the columns, reading from the left to right of the matrix, are structured as shown in Figure C5 :

$$\underbrace{1}_{\text{date}} + \underbrace{n_{ij} + r(C_f^{obs}, C_f^{pred}) + p_C + r(CPUE_f^{obs}, CPUE_f^{pred}) + p_{CPUE}}_{n_f}$$

Figure C5: The structure of `spname_Spatial_Corr.txt` file. The date is written as 'YYYY-mm'. The number n_{ij} is the number of spatial observations for a given fishery f and time step.

¹with a varying from 0 to the maximum age A^+ defined in the model run.

The *C*-code that computes the correlation is based on a script detailed in the book '*Numerical Recipes in C, Press and al. 1994 (14.5 p. 638)*'. If we denote X the observed variable, and Y the estimated one, then at each time step we the spatial correlations are computed as follows:

$$\begin{aligned}\delta_X(i, j) &= X(i, j) - \langle X(i, j) \rangle_{i,j} \\ C_{XY} &= \langle \delta_X \delta_Y \rangle_{i,j} \\ r &= \frac{C_{XY}}{\sqrt{C_{XX} C_{YY}}}\end{aligned}$$

C.2.6 LF_obs.txt

File `spname_LF_obs.txt` stores the observed quarterly and annual length frequency data read by SEAPODYM redistributed to the model age structure and aggregated by fishery and region. Note, if the regional structure is not activated in the parfile, that is `nb_region_sp_B = 0` in the aggregation section, then the regional structure of the LF data is used to write this file, leading to usually a very big file size.

Since both age and regional information is written over quarters, the file contains five table, corresponding to four quarterly and one annual aggregation of the catch-at-age data. The number of rows in each table is equal to the number of age classes and the number of columns depends on the number of fisheries and regions in the model configuration. Each line in these tables is structured as shown in Figure C6:

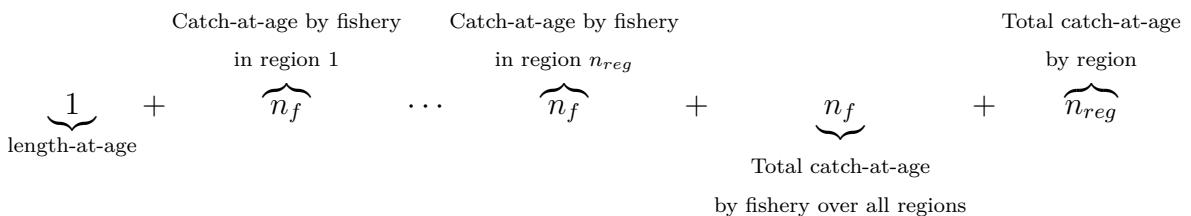


Figure C6: The structure of quarterly or annual LF data in `spname_LF_obs.txt` file. The mean length-at-age is written in cm.

C.2.7 LF_Q_fishery.txt

The data structure in file `spname_LF_Q_fishery.txt` is similar to that in the input file `spname_LF.txt` described in section C.1.2. The difference with the input data file is that

here the LF data are aggregated to the model age classes. The file header is written as follows (example of the skipjack model configuration):

5 15	\leftarrow number of regions number of fisheries
11.65 16.91 21.83 26.43 ...	$\leftarrow \ell_{J+1} \ell_1 \dots \ell_A$
yr qtr mo f reg LF[1] .. LF[11]	\leftarrow column names of the main table

with length-at-age ℓ_p being the mean length (in cm) of individuals in age classes p starting from the first age in the young life stage (Table 1.1). The lines of the main table are structured as shown in Figure C7.

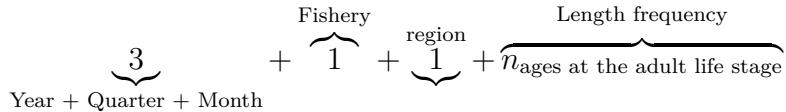


Figure C7: The structure of `spname_LF_Q_fishery.txt` file.

C.2.8 LF_Q_sum.txt

The structure of file `spname_LF_Q_sum.txt` storing the predicted length-frequency statistics is identical to that of file `spname_LF_obs.txt`. See Figure C6 for details.

C.3 DYM files

The binary DYM files are standardized binary files designed specifically to store SEAPODYM input and output variables. These files can be viewed with the GUI software SeapodymView (see D.1) or handled manually using a scripting languages such as R or Python (section D.3). Note, during SEAPODYM development two types of DYM files, DYM1 and DYM2, were implemented, however only DYM2 is currently supported. This section describes DYM2 type.

C.3.1 Structure of a DYM file

All DYM files have the same structure: a header and a single three-dimensional variable written as a sequence of 2D-arrays (matrices) along Z dimension denoting either time or age. Consequently, all input files (section 3.4.4) that store the time evolution of spatial variables in the model configuration must have identical size. Likewise, all SEAPODYM

output files in DYM format (section 3.6.2) have the same size, which may differ from the size of input files or between simulations depending on the specified time period in the simulation.

The header

The DYM header contains information on model domain and dimensions:

DYM_TYPE	{	$(9 + 3 \times (\text{NLON} \times \text{NLAT}) + \text{NLEVEL}) \times 4$
GRID_ID		
MIN_VALUE		
MAX_VALUE		
NLON		
NLAT		
NLEVEL		
START_DATE		
END_DATE		
XLON [NLAT] [NLON]		
YLAT [NLAT] [NLON]		
ZLEVEL [NLEVEL]		
LAND_MASK [NLAT] [NLON]		

where the expression on the right-hand side gives the size of the header in bytes as all values are written as 4-byte objects. Detailed description of each value in the header as well as their types are provided in Table C.3.

Spatial data

The 3D variable is written as a sequence of 2D arrays sorted in chronological order from the first to the last matrix, each matrix being read/written from north to south and east to west (see Figure C8). Contrary to the 4-byte data types in the header, the DYM variables are declared as `double`, i.e, the storage size of 8 bytes (Table C.3). Unfortunately, the current DYM2 format does not store the units of the variable written in the DYM file.

DYM conventions

SEAPODYM has a few conventions that impose the rules for creating and interpreting DYM files. They are listed below.

- DYM files do not take into account implicit boundary cells of numerical model as shown in Figure 2.1 of Chapter 2. Hence the indices of the DYM variable start always with (1,1) in the numerical model code.

Table C.3: Description of DYM file parameters and 3D variable.

Name	Description	Type
Header		
DYM_TYPE	DYM file type, always DYM2	4-byte char[]
GRID_ID	Grid type can have two value: 0 indicates regular uniform grid, 1 – irregular, e.g., stretched grid	4-byte char[]
NLON	Number of grid cells in longitudinal dimension	integer
NLAT	Number of grid cells in latitudinal dimension	integer
XLON	Array of size NLAT × NLON filled with longitudinal coordinates (in degrees) in ascending order	float[][]
YLAT	Array of size NLAT × NLON filled with latitudinal coordinates (in degrees) in descending order	float[][]
NLEVEL	Number of time steps or age classes	integer
START_DATE	First decimal date or age index of the series, for dates see eq. C.3	float
END_DATE	Last decimal date or age index of the series, for dates see eq. C.3	float
ZLEVEL	Vector of decimal dates or indices of age classes	float[]
LAND_MASK	The land mask of dimension NLAT × NLON with values 0 for land and 1 for the sea cells	integer[][]
MIN_VALUE	Minimum value of DYM variable over entire time period	float
MAX_VALUE	Maximum value of DYM variable over entire time period	float
3D variable		
VAR	Data matrices of dimension NLAT × NLON, the units depend on data type	double

- Dimensions NLON and NLAT correspond to n_x (2.2) and n_y (2.3) respectively.
- Coordinates giving the corners of rectangular domain (i.e., the outermost nodes of first and last grid cells) in the XML parameter file relate to DYM dimensions NLON and NLAT as follows:

$$\text{NLON} = (\text{longitudeMax} - \text{longitudeMin}) \frac{60.0}{\Delta X} \quad (\text{C.1})$$

$$\text{NLAT} = (\text{latitudeMax} - \text{latitudeMin}) \frac{60.0}{\Delta Y} \quad (\text{C.2})$$

- The use of Arakawa-A grid implies that the model variable is placed in the centre of the grid cell, therefore coordinates XLON, YLAT in DYM files should give the centre of the grid cell.
- Same as above, the use of Arakawa-A grid implies the variable written in DYM file should be valid at the centre of the grid cell.
- The date in the DYM file is stored in a decimal format. Two conventions exist in SEAPODYM: 1) monthly time-stepping considering 360-days year and 2) standard time-stepping with $\Delta T < 30$, hence considering standard date format. The following conversions from standard date to the decimal date are implemented by SEAPODYM:

$$\text{decimal_date} = \begin{cases} \text{year} + \frac{\text{month}-1}{12} + \frac{\text{day}}{360}, & \Delta T = 30 \\ \text{year} + \frac{\text{day_of_year}}{\text{nb_days_year}}, & \Delta T < 30 \end{cases} \quad (\text{C.3})$$

where `day_of_year` is the sequence day of the year and `nb_days_year` is the number of days in current year. Consequently, any other conversion may lead to improper interpretation of dates and hence potential misuse of matrices written in DYM files.

- The land mask written in DYM file contains only two layers, land and sea, denoted by 0 and 1 respectively. This mask, proper to the DYM variable, should not be confused with the three-layer land mask used in numerical model run (section 3.4.2). Hence, any other mask values written in the DYM file will be ignored by the numerical model and by SeapodymView software (next section).

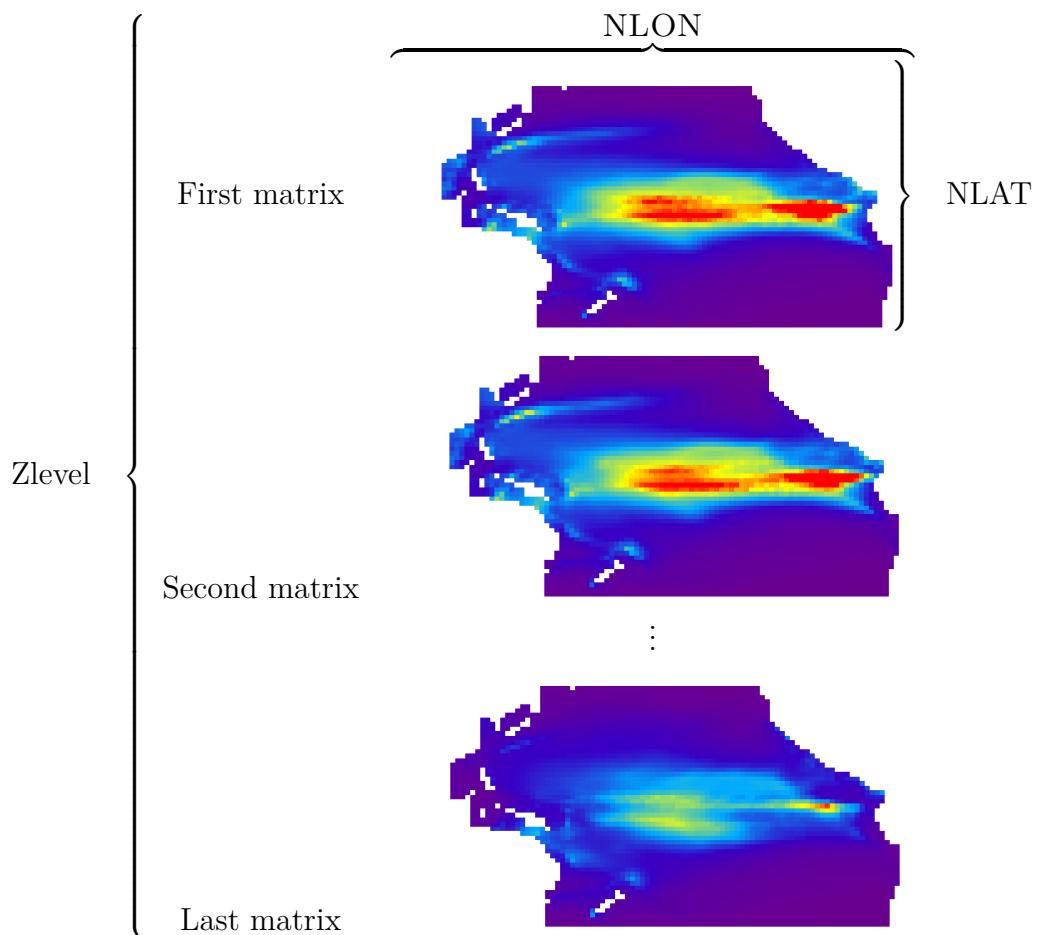


Figure C8: The order and arrangement of a two-dimensional variable in the DYM files containing ZLEVEL 2d-arrays, i.e., matrices of size NLAT×NLON. Each matrix is written by rows moving in latitudinal dimension from north to south, each row includes complete longitudinal vector of values, writted from west to east. This relates to the SEAPODYM convention that the origin (0, 0) is placed in the north-west corner of the rectangular domain.

Appendix D

Toolbox

D.1 SeapodymView

The SeapodymView software was designed at SPC to manipulate and visualize DYM files. This software has a dedicated documentation, therefore here we provide only a short description of essential functionalities of SeapodymView that are used more often, recommending to read its reference manual for the complete information. SeapodymView is a java application, which runs both on Linux and Windows. Note, the language settings depend on the computer set-up, so the names in the menus cited here will be different if default language is not English. Several options for the **Look and feel** of the software are available.

D.1.1 Visualization of DYM variables

Getting started with SeadpodymView is simple and intuitive. Upon loading, the software opens the dialogue window (Figure D1, panel a) suggesting to select DYM file from the home directory (first load), or from the directory that was previously selected. In the right panel of this dialogue window the software shows the information written in the DYM header. If SeapodymView was already open and one file was already displayed, more files can be loaded from menu **File → New**, or by clicking the button **Create a new document**. The software reads the data from selected DYM file and opens the pop-up window displaying its first spatial distribution (Figure D1, panel b).

The map is displayed within the entire latitudinal and longitudinal ranges. It can be zoomed in and out using the mouse wheel. Otherwise the regional zoom can be achieved by selecting the rectangular zone of interest while holding the mouse wheel down. Another click of the mouse wheel on the selected region gets the entire map back. By default, the colorscale for the spatial distribution is confined within the **MIN_VALUE** and **MAX_VALUE** written in the DYM file, which is not always optimal in terms of visualization. So, the minimal and maximal displayed values can be modified via right click of the mouth on the map and selecting **Color**

palette from the pop-up menu. Note also that it is possible to see the values on the map with the left mouse click. The value on the intersection of two perpendicular lines are always displayed in the bottom panel of the window, which shows also the current coordinate and the date.

The spatial distributions can be advanced in time with help of buttons on the left-hand panel (green triangles with vertical bar, see Figure D1, panel b). Another useful functionality of SeapodymView is the animation of spatial distributions, controlled with help of buttons **Play the animation**, **Stop the animation** etc. on the left-hand panel. Besides, if several DYM files are displayed, it is possible to synchronize the animation by placing each spatial distribution at the same time step and linking them using button **Select everything**. The speed of animation can be adjusted, see menu **Properties/Advanced**.

Also, SeapodymView can visualize several DYM variables overlayed on the same map. Thus, it is possible to overlay the fish density biomass with other variables, such as sparse catch data, ocean currents, weekly or monthly individual trajectories. See **Properties** in the pop-up menu activated by the right click of the mouse on the map.

Note that the software allows opening the output DYM file while the simulation is still running. The only condition that is required by SeapodymView is that at least three matrices are written in the DYM file. This functionality can be particularly useful for the long runs, allowing a quick check of the model output without waiting the end of simulation.

D.1.2 Temporal and spatial extractions

SeapodymView can extract DYM variables over the region of interest or over the selected time period (see menu **Tools/Extraction/Seapodym**). Moreover, upon extracting, the variable can be aggregated resulting either in its temporal or spatial average.

Temporal averaging => Spatial matrix. Let's call $\phi(x, y, t)$ the value of a DYM variable at the position (x, y) and at time t . A temporal average can be done as $(\langle \phi(x, y, t) \rangle_t)$, where the notation $\langle \dots \rangle_i$ means an average over the dimension i . As a result we obtain a two-dimensional distribution of the size derived from the variable's resolution and the selected region.

To do that, go to **Tools** → **Extraction** → **Depth averaging**. Then, click on tab **Options** and select **Temporal averaging => Spatial matrix**. Then, go back to the **Selection** tab and adjust the regional coordinates if necessary. Note, the temporal average can either be saved to another DYM (.dym) file, or to the ASCII (.txt) file. The choice of the file type can be done in the same dialogue window.

Spatial averaging => Time series. Here, the following operation is performed with DYM variable: $\phi(t) = \langle \phi(x, y, t) \rangle_{x,y}$, which provides the temporal evolution of the its average over a given region. So, the result is the time series.

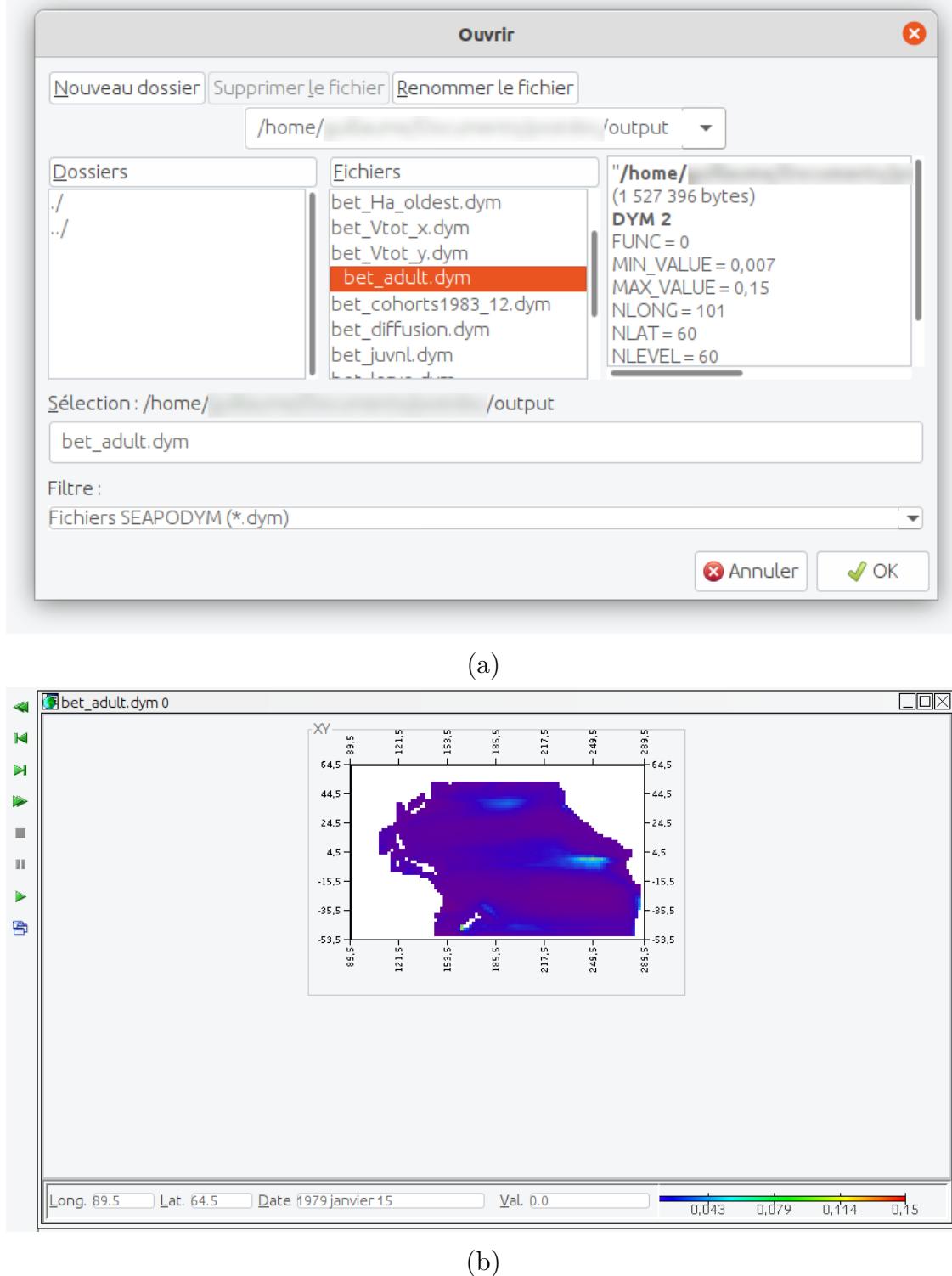


Figure D1: Loading a **.dym** file with Seapodym View. (a) Dialogue window of the loading file, showing the input directory, the list of DYM files and the contents of the file header. (b) File loaded and map displayed and fully extended on the visualization pane. The colorscale 'Temperature' goes from the dark purple to the red for the lowest to highest values being displayed. The buttons on left-size panel allow playing the animation or to advance the spatial distribution in time by one time step.

To do this operation, go to **Tools** → **Extraction** → **Depth averaging**, then in the tab **Options** select **Spatial averaging => Time series**. Go back to the **Selection** tab, and choose the space-time domain to compute average and extract the time series. The output will automatically be saved into an **ASCII** file.

D.1.3 Arithmetic operations

SeapodymView allows performing basic arithmetic operations with two or more DYM files. To do that, click on **Tools** → **Math Operation**. The first file has to be selected in the upper panel named **Input file**. Then the user is suggested to choose the **Operator** among implemented addition, subtraction, multiplication, division and power). The variable from selected DYM file can either be multiplied (subtracted...) by the **Coefficient**, or the operation can be performed on several DYM variables. In the latter case, activate the option **Use the sum of these auxiliary files**, and the dialogue window will allow selecting one or several more DYM files. The result of the operation is stored in the new DYM file, its name and location should be specified in the same dialogue before the **Ok** button gets active. Once the prompt message informing on the file creation has popped up, the new file can be loaded and the result of the operation can be viewed.

D.1.4 From DYM to ASCII file

This conversion might be useful to perform quick quantitative analysis or to plot DYM variable or its regional/temporal extractions using high-level programming languages such as **R**, **Python**, or **Matlab**.

To extract the variables into a text file, go to **Tools** → **Extraction** → **Matrix**. Check the path directory. The name will be set automatically depending on the format chosen to store in the matrix. The file **Txt** type is selected automatically. Then in the **Selection** tab choose one of the five variables for extraction. No adjustment is necessary if choosing the spatial coordinates (**XLON**, **YLAT**) or time (**ZLEVEL**) vector. However, when selecting the land mask **Msksp** or the DYM variable (**Data - main data block**), it is possible to specify the region and the time period for extraction. Clicking **Ok** generates the output ASCII file in the selected folder.

D.2 Grid and Mask Builder

To build the grid for the chosen geographic area and to create the multi-layered land mask we suggest to use the software called GMB (Grid and Mask Builder). It allows also manipulation of land mask such as modification of domain boundaries, closure or opening of sea cells,

filtering out the small areas (lakes) surrounded by land and so on. Using ETOPO2 (2-minute gridded global relief data, see information on <http://www.ngdc.noaa.gov/mgg/global>) topography maps, GMB allows building regular and stretched grids and creating land masks and masks for arbitrary chosen depth level. Note that for the moment we consider regular grids only, i.e., orthogonal grids with constant resolution in latitude and longitude, although the mixed-resolution type grids (e.g. ORCA) might be envisaged for the future releases of the SEAPODYM application with parameter estimation.

GMB was created and further developed as a plugin for SEAPODYM, which accomplishes grid generation, land mask construction and data interpolation on designed grids. For interpolation of input environmental data such as ocean currents, temperature, oxygen and primary production onto new grids, GMB includes a cubic splines interpolation routine and for transferring sparse catch data averaged over square cells there is the routine for redistributing data within new grid cells.

After downloading software create three directories in the folder GMB: `maps`, `output` and `data`. In order to start working with GMB, you need to have ETOPO2 maps and contour files. If you didn't download them with GMB, you have two options: either extract map directly from NOAA website [Geodas](#) or install GEODAS grid translator from [.](#) Current version of GMB reads binary files with GRD98 header (extension g98), data should be written in little-endian, 4-byte integer format. All these settings should be chosen while retrieving map interactively from website or from installed GEODAS software.

To run GMB under Windows or Linux, one must have Java Runtime Environment installed (for latest version go to website [.](#)). In Windows, click on jar file in order to run the software. If you are going to work with large, high resolution topography maps, it's better to allocate more memory by typing command from the current GMB directory:

```
C:\WINNT\system32\java.exe -Xmx512m -jar GMB.jar
```

in Command Prompt window or simply create shortcut for GMB and indicate it in a Target field. For example,

```
C:\WINNT\system32\java.exe -Xmx512m -jar "C:\Seapodym\GMB\GMB.jar"
```

If you run GMB under Linux, type in terminal

```
java -jar GMB.jar
```

from directory with jar file. Read GMB help (or directly `readme.txt`) for more details on the functionalities implemented in this software.

For example, to create a 1° grid for Pacific Ocean domain as shown in Fig D2 and 3-layer mask using GMB, extract the map for the area of interest with ETOPO topographic data (Fig D2), follow the steps: choose the following menu commands (GMB user's menu activates

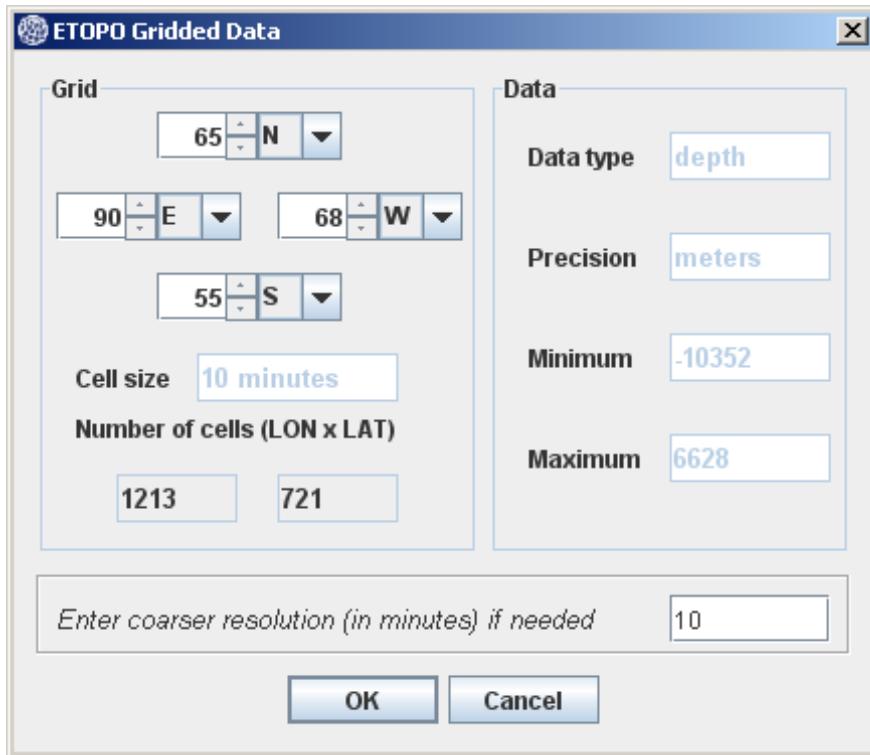


Figure D2: Loading the map of Pacific Ocean.

successively) or click on the corresponding buttons: 1) Load Map; 2) Build Grid; 3) Create Mask; 4) Mask/Apply Boundary Filter. Available Etopo maps with different resolution are stored in the directory `maps`, the files having extension `'.g98'`. While extracting geographic area and building grid remember that approximation scheme of PDEs in SEAPODYM is based on Arakawa A grid, so first, the domain coordinates must be extended by half of a desired grid resolution, i.e to `longitudeMin-.5Δx`, `latitudeMax-.5Δy` and so forth; and second, the number of longitudinal and latitudinal grid nodes must be $n_x + 1$ and $n_y + 1$, where $n_x \times n_y$ is the dimension of all model variables. The decision on the model grid resolution should rely on the following information: 1) the dimension of available physical and biogeochemical variables, either predicted with models (OGCM, NPZD) or observed, 2) the size of the grid cells should be sufficiently small to resolve studied dynamical features, 3) the resolution of available geo-referenced fishing data.

Note that land mask created directly from topography data can be different from the one being used in OGCM or NPZD models, that is why it is recommended to use the data mask (mask, extracted from oceanographic and biogeochemical data) as an initial land mask (use menu `File/Load mask...`). Adding more layers to the mask, e.g., based on the depth defining epipelagic and mesopelagic layers, can be done in GMB as well (use menu `Mask/Mask properties...` to set the depth and mask flag, then clicking "create mask")

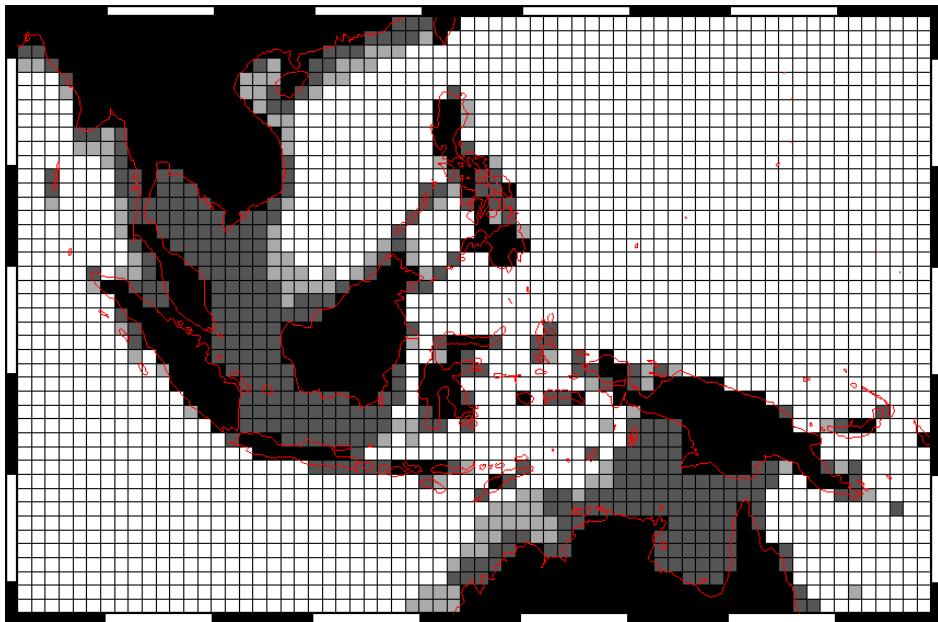


Figure D3: Indo-Pacific regional domain with 1° grid and 3-layer mask.

will add another layer with specified properties). However, this method will work only in case if depth of the layer is considered to be constant. For the definition of variable layer depths a bit more complex procedure must be undertaken (see below). Since the resulting land mask defines the complex boundary of computational domain, an important step to make before saving the mask file, is to mask existing near-border cells which will cause the biomass leak in the finite-difference approximation scheme of ADR equations. Leak occurs when one computational cell is surrounded by three land cells (or if the ocean cells create one-cell channel), as such layout of grid cells makes it problematic to create reflecting boundary condition. The GMB software filters such boundary cells in one click (menu **Mask/Apply boundary filter...**). Finally, the mask is saved by GMB into a simple text file with the table of numbers, masking land with 0 and ocean (computational) cells with 1, 2 and 3 etc. according to the number of vertical layers specified for the simulation (see Fig. D3).

D.3 R scripts

Various R functions were written in order to facilitate data preparation as well as model analyses and validation in SEAPODYM. Here we provide descriptions of only a few selected routines that are essential in extraction of data from SEAPODYM DYM and ASCII files, plotting outputs on a map and preparing fisheries data for simulations with or without scenarios. These functions are part of the following libraries:

```
dym-works/
  read_varDYM.R
  write_varDYM.R
```

The example of use of these read-write functions for DYM files

```
res<-read.var.dym(file.in,t0,tfin,region,deltaT)
```

where `t0` and `tfin` are the combinations of two or three values (`year,month,day`), with `day` which can be omitted for monthly time stepping (`deltaT=30` by default). The returned value of this function, i.e. variable `res` is a `list(x,y,dates,var,landmask)` with the components that are necessary to manipulate DYM data: vectors of coordinates and dates, a variable `var`, extracted over region defined in the corresponding argument and a land mask. There are other useful functions such as `read.txy.dym` to read only dimensions from a given DYM file, and `read.restart.dym` to read the restart file of SEAPODYM. The syntax for the function to write DYM file is

```
write.var.dym(file.out,dates,x,y,landmask,var)
```

This function does not return a value, instead the SEAPODYM variable `data.out` is written to a DYM file a `file.out` with all dimensions and a land mask. Another library `fisheries` contains R functions designed to work with SEAPODYM fisheries data files (effort and catch, see [C.1.1](#)), namely, to read, write fisheries files, compute monthly climatology, introduce effort multipliers, plot final distributions of climatological catch data :

```
fisheries/
  call_EC_climatology_func.R
  libfd.R
  make_EC_climatology.R
  rw_sea_fisheries.R
```

The graphical outputs with fisheries data as well as of extracted DYM variables can be produced with the functions `palettes` and `nice.map` that are implemented in the following files:

```
maps/
  Palette.R
  plotmaps-GMT.R
```

The syntax for `nice.map` function, that will plot the geographical map of the selected region

is as follows

```
  nice.map(x,y,resolution=5,grid=FALSE,...)
```

where `x` and `y` are simple increasing values of coordinates, provided as, e.g. `x=100:290, y=-10:10` for tropical Pacific, or `x=-70:20, y=-10:10` for tropical Atlantic. Then, the extracted `var` can be overlaid on the previously plotted map using generic R functions, e.g. `image(x,y,var)`. Note that functions in `plotmaps-GMT.R` use the Generic Mapping Tools (GMT) libraries, therefore before using them check that GMT is installed on your computer. If not, visit page <http://gmt.soest.hawaii.edu/projects/gmt/wiki/BuildingGMT> for installation instructions.

The installation and the use of these libraries requires two environment variables to be declared in the user's `.bashrc` file:

- 1 `SEA_R_HOME`: the path to the folder with all above R scripts
- 2 `SEA_HOME`: the main SEAPODYM directory, with folders called `data` (all forcings placed here), `run` (model runs with parameter files and outputs) and `fisheries` (fisheries data files).