

GoFish: A Next-Generation ToolKit for Modeling Migratory Fish and Quantifying Environmental Risk in Estuaries

Vanessa Quintana

2025-11-23

Contents

Preface	9
Motivation	9
Application Context	9
How to Use This Library	10
1 Introduction	11
1.1 Structure	11
2 Agent-Based Models	13
3 Co-Development of ABMs: Penobscot Estuary Community Workshop August 6, 2025	17
3.1 Core Co-Development Themes	17
3.2 Workshop Development	18
3.3 Facilitating the Workshop	21
3.4 After the Workshop	22
4 Metabolism	27
4.1 Overview	27
4.2 Purpose	27
4.3 Entities, State Variables, and Scales	27
4.4 Process Overview and Scheduling	29
4.5 Design Concepts	29
4.6 Initialization	30

4.7	Submodels	31
4.8	Netlogo Implementation	33
5	Digestion	37
5.1	Overview	37
5.2	Purpose	37
5.3	Entities, State Variables, and Scales	37
5.4	Process Overview and Scheduling	39
5.5	Design Concepts	39
5.6	Initialization	40
5.7	Submodels	41
5.8	Netlogo Implementation	42
6	Salinity Exposure	45
6.1	Overview	45
6.2	Purpose	45
6.3	Entities, State Variables, and Scales	45
6.4	Process Overview and Scheduling	47
6.5	Design Concepts	47
6.6	Initialization	48
6.7	Submodels	48
7	Migration Cues	57
7.1	Overview	57
7.2	Purpose	58
7.3	Entities, State Variables, and Scales	58
7.4	Process Overview and Scheduling	59
7.5	Design Concepts	61
7.6	Initialization	61
7.7	Submodels	62
7.8	1. Identifying Cue Days (Preprocessed in R)	62
7.9	2. Alewife Migration Probability	62
7.10	Netlogo Implementation	62

<i>CONTENTS</i>	5
8 Contaminant Exposure	65
8.1 Overview	65
8.2 Purpose	65
8.3 Entities, State Variables, and Scales	65
8.4 Process Overview and Scheduling	67
8.5 Design Concepts	68
8.6 Initialization	68
8.7 Submodels: Mercury Exposure and Uptake	69
8.8 Submodels: Methylmercury Exposure and Uptake	69
9 Filter Feeding	75
9.1 Overview	75
9.2 Purpose and Patterns	75
9.3 Entities, State Variables, and Scales	76
9.4 Process Overview and Scheduling	78
9.5 Design Concepts	78
9.6 Initialization	79
9.7 Submodels	80
9.8 Overview	84
9.9 Purpose	85
9.10 Entities, State Variables, and Scales	85
9.11 Process Overview and Scheduling	86
9.12 Design Concepts	87
9.13 Initialization	87
9.14 Submodels	88
9.15 Netlogo Implementation	89
10 Landward Migration	93
10.1 Overview	93
10.2 Purpose	93
10.3 Entities, State Variables, and Scales	93
10.4 Process Overview and Scheduling	95

10.5 Design Concepts	95
10.6 Initialization	95
10.7 Submodels	96
11 Seaward Migration	107
11.1 Overview	107
11.2 Purpose	107
11.3 Entities, State Variables, and Scales	107
11.4 Process Overview and Scheduling	108
11.5 Design Concepts	109
11.6 Initialization	109
11.7 Submodels	110
11.8 NetLogo Implementation	112
12 Schooling	119
12.1 Overview	119
12.2 Purpose	119
12.3 Entities, State Variables, and Scales	120
12.4 Process Overview and Scheduling	120
12.5 Design Concepts	121
12.6 Initialization	121
12.7 Submodels	122
12.8 Netlogo Implementation	123
13 Selective Tidal Stream Transport	127
13.1 Overview	127
13.2 Purpose	127
13.3 Entities, State Variables, and Scales	128
13.4 Process Overview and Scheduling	129
13.5 Design Concepts	129
13.6 Initialization	129
13.7 Submodels	130
13.8 Netlogo Implementation	132

14 Predation and Fleeing Behavior	137
14.1 Overview	137
14.2 Purpose and Patterns	137
14.3 Entities, State Variables, and Scales	138
14.4 Process Overview and Scheduling	141
14.5 Design Concepts	142
14.6 Initialization	143
14.7 Submodels	144
15 Model Building Tutorial: Staging & Schooling	149
15.1 Implementation in NetLogo:	149
15.2 Implementation in R:	150
16 Model Building Tutorial: Landward Migration & Selective Tidal Stream Transport	151
16.1 Implementation in NetLogo:	152
16.2 Implementation in R:	152
17 Complex Model Building Tutorial: One-Way Migration	153
17.1 Implementation in NetLogo:	153
17.2 Implementation in R:	154
18 Modeling Toolkit	155
18.1 Learning Resources	155
18.2 Best Practices	155
18.3 Code Repositories & Examples	156
References	157
Glossary	159
A	159
B	160
C	160
D	161

E	161
F	162
H	162
I	162
L	163
M	163
O	163
P	164
S	164
T	166

Preface

This library provides a comprehensive, modular framework for developing and documenting agent-based models (ABMs) that simulate the movement, behavior, and environmental interactions of migratory fish in coastal aquatic systems. It is designed to support the standardization and implementation of ABMs in fisheries management, enabling researchers and practitioners to address complex environmental questions and evaluate remediation or restoration scenarios.

Motivation

The goal of this resource is to support students, researchers, and decision-makers by making agent-based modeling of migratory fish more accessible, reproducible, and applicable to real-world fisheries and habitat management challenges. By providing a standardized modular framework for key behavioral processes, this library promotes consistency, transparency, and credibility in ecological forecasting and decision support tools. It also establishes a foundation for critical conversations about the behaviors and functions represented in modeling, while supporting the empirical quantification of ecological relationships that influence movement, survival, and habitat use of migratory fish.

Application Context

This library was originally developed in support of research on the influence of tidal behavior and contaminant exposure on anadromous fish in the Penobscot River Estuary. However, its modular design allows for application to other estuarine and coastal systems where fish respond to environmental changes (i.e., salinity, velocity, and pollutants).

How to Use This Library

Each function or behavior in this library can be combined with others to build a complete agent-based model for anadromous fish. These functions are designed to be adaptable, and easily configured for different species, life stages, or site-specific conditions.

*For questions, feedback, guidance on implementation, or **interest in adding to the library**, please contact **Vanessa Quintana** at **ma-han.vanessa98@gmail.com**.

Chapter 1

Introduction

Anadromous fish species such as river herring, striped bass, salmon, and sturgeon navigate coastal and estuarine systems that are increasingly affected by human activity, climate change, and legacy contaminants. Modeling their movement and behavior at fine spatial and temporal scales requires tools that can integrate physiological stressors, environmental variability, and behavior-based decision-making.

Agent-based models are among the most powerful tools available for ecological forecasting and fisheries management, but they are also among the most complex. Their structure and computational demands can make them difficult to apply in practical management settings. Many biologists and ecologists who hold deep, species-specific expertise often have limited training in advanced programming or systems modeling.

As a result, traditional approaches to simulating anadromous fish frequently oversimplify or exclude key biological functions and environmental interactions such as salinity exposure and contaminant exposure. Existing models also lack standardized representations of these behaviors and interactions, limiting the applicability of results and reducing their usefulness for applied management. This function library was developed to address these limitations by offering modular, empirically grounded components designed for use in modeling migratory fish. Each function is clearly documented and can be applied independently, allowing for transparent testing, modification, and reuse across a wide range of ecological scenarios and sites.

1.1 Structure

Each chapter in this library corresponds to a major behavior, physiological function, or foundational component relevant to migratory fish, beginning with:

- **Co-Development of Agent-Based Models**
- **Calculate Metabolism**
- **Salinity Exposure**
- **Temperature Exposure**
- **Contaminant Exposure**
- **Digestion**
- **Filter Feeding**
- **Lipid Catabolism**
- **Landward Migration**
- **Seaward Migration**
- **Schooling**
- **Selective Tidal Stream Transport**
- **Predator-prey interactions**

The final chapters provide guidance on how to integrate multiple functions into a complete agent-based model, demonstrating how these components work together to simulate fish behavior in dynamic coastal and estuarine systems.

Within each chapter, each function or behavior is documented using the ODD protocol [Grimm et al., c,a,b]. The ODD (Overview, Design concepts, Details) protocol is a standardized framework for describing agent-based models. It promotes transparency in model development and ensures consistency across implementations, especially when integrating multiple behavioral or ecological functions.

- **Overview** provides the purpose of the model component, identifies the entities involved (e.g., fish agents, environmental patches), and outlines the general processes.
- **Design** concepts describe the key theoretical underpinnings such as emergence, adaptation, objectives, sensing, stochasticity, and interaction.
- **Details** specify initialization steps, input data requirements, and the rules or submodels that govern behavior.

By following the ODD protocol, this library ensures that each function is self-contained, interpretable, and ready for adaptation to a wide range of species, sites, or management scenarios.

Chapter 2

Agent-Based Models

Agent-based models (ABMs) simulate the behavior of individual organisms—called “agents”—and how their interactions with each other and with their environment create larger-scale patterns. Each agent follows a set of simple rules based on physiology, behavior, or environmental cues. When many agents follow these rules simultaneously, complex system-level outcomes emerge, such as migration pathways, schooling patterns, habitat bottlenecks, or exposure hotspots. ABMs are especially useful for ecological systems because they allow researchers and communities to explore how individual decisions scale up to influence population dynamics, habitat use, and risk in changing environments.

ABMs are designed to capture processes that depend on **behavior, timing, environmental variability**, and **individual differences**. They allow modelers to explore how organisms respond to their surroundings, make movement decisions, and interact with other individuals or species. Because agents experience the environment at fine spatial and temporal scales, ABMs can reveal patterns that occur only when many individuals interact across a dynamic landscape.

2.0.1 What ABMs Can Help Us Understand

ABMs can answer questions related to:

- How individual movement decisions accumulate into migration routes or staging locations
- When and where organisms are most exposed to environmental risks
- How behaviors such as foraging, schooling, predator avoidance, resting, or spawning influence exposure or survival

- How variation in traits (size, energy, age, or species) contributes to different outcomes
- How environmental conditions shape habitat use, movement choices, or interactions between species
- How system-level patterns emerge from local behaviors and simple rules
- How alternative conditions or restoration actions might alter movement, habitat use, or exposure

These models are especially useful when management questions depend on **patterns created by individual behavior**, rather than population totals alone.

2.0.2 What ABMs Are Not Designed to Do

ABMs cannot provide or replace:

- Exact estimates of population abundance
- Precise contaminant concentrations inside individual organisms
- Direct measurements of chemical or physiological processes
- Predictions of exact numbers of predation or spawning events
- Large-scale demographic projections without additional modeling components
- Replacements for field sampling, monitoring programs, or laboratory measurements

Instead, ABMs offer **process-based insight**, showing how behavior and environmental conditions interact to create observable patterns.

2.0.3 Emergent Behaviors

Emergent behavior refers to patterns that arise when many individual agents follow their own rules, creating outcomes that cannot be predicted from observing a single fish on its own. These outcomes develop only when many individuals interact with each other and respond to their surrounding environment at the same time. Many ecological processes such as migration timing, habitat bottlenecks, exposure hotspots, schooling dynamics, predator and prey encounters, and shifts in group structure are emergent rather than linear. They form through small, individual decisions that accumulate into larger and sometimes unexpected system-level patterns.

By visualizing and simulating emergence, ABMs make it possible to see how these individual behaviors scale up to influence population level outcomes and habitat use patterns. This helps identify the specific conditions, behaviors, and locations that shape system-wide responses. For management, this is valuable because it shows how small-scale decisions made by individual fish can contribute to broader patterns that influence risk, habitat quality, restoration effectiveness, management actions, and ecological resilience. Understanding emergence allows managers and communities to explore why certain patterns occur, identify potential leverage points, and anticipate how changes in environmental conditions may alter system behavior in the future.

Chapter 3

Co-Development of ABMs: Penobscot Estuary Community Workshop August 6, 2025

This page documents how a participatory workshop informed the functions within this coding library, the representation of physiological processes, behaviors of migratory fish, and their interactions within an estuary. The purpose of this page is to showcase the value added by co-development and increase transparency about the development of this coding library.

The Penobscot Estuary Community Modeling Workshop was designed from the beginning as a co-development space for an agent based model to address mercury toxicity dynamics of anadromous fish within the estuary. The day invited Tribal collaborators, community members, managers, and researchers to shape the modeling objectives, including which species, processes, behaviors, and outputs the model should represent, and how those outputs should be represented in relation to contaminant exposure in the Penobscot River Estuary.

3.1 Core Co-Development Themes

Several themes guided the design of the workshop and the way it connects to model development.

- **Shared problem framing.** The workshop centered a shared concern about mercury contamination in the Penobscot River and its effects on

sea-run fish, subsistence access, and recreation use, rather than starting from a predefined modeling objective.

- **Attention to both internal and external processes.** Breakout discussions were split into two rounds, one for “internal” or biological functions of fish such as osmoregulation, thermoregulation, bioaccumulation, and resting, and the second round focused on “external processes” or external interactions with intra- or interspecific factors and environmental responses such as swimming, predation, foraging, and spawning. This structure maps directly onto modules detailed within this library.
- **Participant driven priorities.** Participants were asked which species were most important to understand for management of the system or species that should be prioritized based on community interest, which functions and interactions they believed to be relevant for understanding mercury toxicity in migratory fish, and how those behaviors should be represented.
- **Outputs that answer real questions.** Each participant had the opportunity to propose a question or hypothesis to test using the developed functions and select preferred outputs from the functions such as migration paths, exposure metrics, or habitat use. These wish-list requests now correspond to specific model output routines and visualizations.
- **Accessible language and shared vocabulary.** A glossary of terms translated physiological, hydrodynamic, and modeling concepts into accessible language. This glossary can be found at the end of this document and is intended to support a range of readers.
- **Iterative relationships, not a single event.** The workshop was framed as one step in an ongoing partnership. Feedback forms, follow-up contact options, and explicit invitations for later review are built into the process.

3.2 Workshop Development

3.2.1 Pre-Workshop Collaboration Steps

Planning the workshop involved months of coordination with partners, including the Penobscot Nation, faculty and staff at the University of Maine, and federal collaborators. The planning process itself was a first stage of co-development.

Key steps included:

- **Clarifying objectives with partners.** Early conversations identified contamination in the Penobscot River, migratory fish, and access to safe subsistence resources as the shared focus. Partners helped guide agenda

development, identify potential participants of interest, and identify community outputs of interest to support Tribal management. Their suggestions were used when conducting outreach.

- **Aligning model scope with community priorities.** The agenda was constructed so that introductory talks on Penobscot Nation natural resource management and contaminants of interest, an overview of estuary dynamics (connection of fish + physical processes + contamination), and an introduction to agent based modeling were followed by interactive surveys and break-out discussions. This allowed participants to identify anadromous species, behaviors, and processes of interest, which then feed directly into the model design.
- **Preparing interactive polls and prompts.** A Mentimeter survey was developed to help participants identify priority species, important environmental drivers, and fish behaviors that shape exposure. These questions now map to the list of behaviors and parameters implemented in the model.

Question Example

- **Designing materials for many kinds of learners.** Each participant received a notepad and workshop folder, which included a plain-language glossary, a participant agenda, a hypothesis and output worksheet, and a feedback form. These materials supported participants who learn best through reading, writing, discussion, or visual exploration. Break-out discussions were accompanied by visual demonstrations of function demos, allowing participants to directly address draft model functions for each behavior.

Find examples below.

Note: these are presentations of functions before participant feedback. Updated functions in this library reflect changes informed by the workshop.

3.2.2 Identifying Participants of Interest

Participant selection was guided by the goal of gathering diverse knowledge about the river and its fish.

- **Tribal Collaborators.** Representatives from the Penobscot Nation were invited to collaborate because the Penobscot River Estuary is the Ancestral Homeland of the Penobscot Nation and other Wabanaki Peoples. It was necessary to prioritize their concerns and support their management. We wanted to give them a platform to share their work and management

perspectives on migratory fish and contamination. Indigenous people have sovereign rights to migratory species, which have been deeply impacted by contamination in the estuary.

- **Community Experts.** The workshop brought in local managers, fishermen, researchers, engineers, and students whose work focuses on estuarine dynamics, fisheries, remediation, and environmental management in the Penobscot, or who focus on anadromous fish.

3.2.3 Enticing Participants to Attend

The invitations and overview materials emphasized that this was a hands-on, collaborative event where participant input would shape real model structure and outputs—not a passive lecture.

Copy of Invitation

Strategies included:

- **Clear framing.** The overview highlighted how tidal dynamics, fish behavior, and mercury contamination intersect in the Penobscot River, and described specific opportunities for participants to contribute knowledge during breakout sessions and wish-list activities. The invitation also emphasized that participants did not need any modeling experience—only their expertise and perspectives.
- **Concrete outcomes.** Participants could request a specific model output or test a hypothesis relevant to their interests, with results attempted to be shared after the workshop (as feasible within model limitations).
- **Relationship building.** The agenda previewed time for partnership discussions, highlighting opportunities to support Tribal priorities and future collaboration on management and research questions.

3.2.4 Selecting the Venue

The workshop was held at the Innovation Media Research Center at the University of Maine, a space that could support immersive presentations and movement between small and large group activities.

The venue allowed for:

- **A theater-style room with a large screen.** The introductory talks and interactive survey used a dark background, low lighting, and a large screen to create an immersive experience similar to an intimate concert hall, while still leaving space for questions and discussion.

- **A separate interactive space.** Round tables, easel pads, printed hand-outs, and discussion areas were set up in a second room for the world-café breakout sessions, allowing participants to interact and face one another in a conversation-style setting as they discussed each function.

3.3 Facilitating the Workshop

3.3.1 Agenda Activities

The agenda combined presentations, interactive polls, world-café discussions, and reflection.

Key components included:

- **Welcome, blessing, and introductions.** The day opened with a welcome and words from the Penobscot Nation, followed by introductions and an icebreaker activity asking participants why they attended and what they hoped to gain.
- **Map-based activities.** Participants could pin locations on a system-scale map, connecting their lived experience and knowledge to specific reaches of the river and estuary. This place-based interaction showcased lived knowledge and gave participants autonomy over the spatial representation of the system.
- **Penobscot Nation history and river overview.** A management representative of the Penobscot Nation, Dan Kircheis, provided management context for contaminants and migratory fish, grounding the model in current work conducted within the system.
- **Modeling overview and agent based model presentation.** Modeling presentations introduced the Penobscot system, key hydrodynamic and contaminant processes, and the rationale for an agent based approach. Examples of emergent behaviors and validation were shown on a large screen to make model dynamics visible and engaging.
- **Mentimeter priority setting.** Participants completed an interactive survey about priority species, important environmental drivers, and key behaviors. These responses were summarized live and used to frame the breakout sessions.
- **World café on internal fish processes.** Small rotating groups discussed processes such as osmoregulation, thermoregulation, bioaccumulation, and resting. Discussion leaders presented demos and recorded notes, which later informed how these processes are parameterized and sequenced within this library.

- **World café on external behaviors and conditions.** A second set of rotations focused on external functions such as foraging, migration, predation, and spawning. These conversations helped identify how fish interact with their environment and where and when the model should track exposure risk.
- **Wish list outputs.** Participants wrote down specific outputs they wanted from the model and the reasoning behind each request, including migration timing, exposure patterns, energy budgets, and habitat use. The goal is to incorporate as many of these outputs as possible to support subsequent management and research.

Hypothesis Form

- **Model demo.** A short demonstration of the model interface supported visual learners and connected the function modules to emergent behaviors.
- **Partnership and next steps discussion.** The workshop closed with a short session on building partnerships and identifying community projects where the model could support ongoing work. This included relationship-building between participants and completion of a follow-up survey.

Participant Survey

- **Facilitation.** Facilitation roles were shared across a team of organizers and volunteers who kept time, prompted discussion, recorded notes, and supported logistics, which allowed the lead modeler to focus on listening and translating insights into model design.

3.4 After the Workshop

3.4.1 Incorporating Feedback

Immediately following the workshop, participants were invited to fill out feedback forms on what they found most useful, what felt unclear or missing, and how future workshops or model development activities could be improved.

Feedback is being used to:

- Refine explanations of agent based modeling and estuarine processes in the model interface and supporting documents, using more examples, more visuals, and clearer connections between field observations and model rules.
- Adjust the balance between presentation time and discussion time.

- Clarify how participant input has been integrated, so that people can see their contributions in the model structure and outputs rather than only in summary notes.

3.4.2 Post-Workshop Follow-Up

Several tools support ongoing co-development after the workshop day itself. This work is still ongoing as products are completed and co-development of final materials continues.

- **Hypothesis and output forms.** The participant hypothesis and output form captured individual questions, preferred outputs, and desired scenarios. These forms now guide scenario design and output routines in the model (as much as feasible within model limitations and scope), and follow-up emails will share results with those who opted in.
- **Ongoing communication.** Participants who requested follow-up are being contacted with model updates, refinements, and opportunities to review model structure.
- **Follow-up survey.** The feedback form includes an option to receive a later survey on model results, which will allow participants to review outputs, comment on whether they find them credible and useful, and suggest further refinements.
- **Tech Transfer Webinar.** A planned webinar will support technology transfer of the completed model and products to workshop participants, allowing them to see how their input made it into the final product.
- **Integration into model documentation.** This co-development page, the glossary, and workshop-derived figures will be hosted alongside the developed functions to make the participatory process visible to anyone using or reviewing the simulations.

3.4.3 Acknowledgements

The workshop and this co developed model were made possible by the time, knowledge, and care shared by workshop participants and organizers.

3.4.3.1 Participants

Ernie Atkinson
Andrea Casey
Devon Gleason

Heather Hamlin
John Hildebrand
Dan Kircheis
Dianne Kopec
John Kocik
Dan Kusnierz
Chuck Loring
Dan McCaw
Amelia McAvoy
Sasha Milsky
Elias Pinilla
Darren Ranco
Angie Reed
Lauren Ross
Justin Stevens
Joe Zydlewski

3.4.3.2 Volunteers

Nicolas Cyr
University of Maine

Katherine Daza
University of Maine

Joseph Dello Russo
University of Maine

Nalika Lakmali
University of Maine

Saba Molaei
University of Maine

Cristian Rojas
University of Maine

Malavika Sudhakaran
University of Maine

Kaylyn Zipp
University of Maine

3.4.3.3 Organizers:

Vanessa Quintana
University of Maine; USACE Engineering Research and Development Center

Gayle Zydlewski

University of Maine; NOAA Sea Grant

Katrina Armstrong

University of Maine

Todd Swannack

Texas State University; USACE Engineering Research and Development Center

Chapter 4

Metabolism

4.1 Overview

Metabolism determines the energy required for fish to maintain basic physiological function, support movement, regulate ions, digest food, and sustain long term migration. This submodel computes metabolic rate based on size, temperature, age scaling, and allometric relationships. Metabolic efficiency, based on divergence from an agent's optimal temperature, is then scaled toward multiple internal processes including swimming, osmoregulation, and lipid catabolism.

4.2 Purpose

To simulate energetically realistic metabolic costs for migratory fishes in coastal and estuarine environments by accounting for size scaling, temperature dependence, age effects, and dynamic energy partitioning across physiological processes.

4.3 Entities, State Variables, and Scales

4.3.1 Spatial and Temporal Scales

- **Spatial Unit:** Patch (3 m x 3 m resolution)
- **Temporal Unit:** 5-minute time steps (`tick`)

4.3.2 Global Variables

Global Variable	Definition
Q10	Temperature coefficient used for metabolic scaling.
base-constant	Global constant for baseline metabolic scaling (0.1).
age-sensitivity	Scalar controlling how metabolism changes with age (0.5).

4.3.3 Patch Variables

Variable Name	Definition
temperature	The water temperature of the current patch used to calculate temperature dependent metabolic rate.

4.3.4 Agent Variables

Variable Name	Definition
size	Body mass or size index used for allometric scaling.
age-num	Age used in scaling metabolic rate.
optimal-temperature	Preferred temperature at which baseline metabolism is defined.
max-survival-temp	Maximum temperature at which the agent can survive and function.
Met-base	Baseline metabolic constant calculated during initialization.
metabolism-rate	Metabolic rate for the current tick.
total-metabolism	Accumulated metabolism over time.
swim-efficiency	Efficiency applied to movement when metabolic rate is lower or higher than baseline.

Variable Name	Definition
osmoregulation-efficiency	Efficiency applied to osmoregulation when metabolic rate is lower or higher than baseline.
lipid-catabolism-efficiency	Efficiency applied to lipid catabolism when metabolic rate is lower or higher than baseline.
swim-base	Baseline allocation of metabolism toward movement.
E-base	Base metabolic maintenance cost.
E-creation	Energy cost for creating new chloride cells.
E-osmo	Total energy spent on osmoregulation.
motion-ratio	Fraction of metabolism allocated to swimming.
osmo-ratio	Fraction of metabolism allocated to osmoregulation.
M-max	Maximum size for scaling across the population.
max-age	Maximum age for scaling across the population.

4.4 Process Overview and Scheduling

1. Retrieve patch temperature and verify safe environmental inputs.
2. Compute allometric, temperature, and age scalars.
3. Calculate metabolic rate for the current tick.
4. Update metabolic accumulation.
5. Compute a reference baseline metabolic rate.
6. Apply efficiency multipliers to swimming, digestion, osmoregulation, and lipid catabolism.

4.5 Design Concepts

Basic Principles:

Metabolic rate combines allometric scaling, temperature responses, and age scaling consistent with metabolic ecology theory. The model uses a Q10 temperature function, Kleiber type allometry, and age dependent reduction in metabolic intensity.

Emergence:

Energetic constraints emerge from interactions among size, temperature, and age. These constraints then shape movement, digestion, and osmoregulation.

Adaptation:

Agents adapt by adjusting internal efficiencies that influence downstream behavioral and physiological submodels.

Objectives:

Agents maintain adequate energy for survival and migration and balance energetic cost across processes.

Sensing:

Agents sense the temperature of the patch they occupy.

Stochasticity:

The function itself is deterministic but stochastic variation in individual size and age creates emergent metabolic variation.

Collectives:

Scaling uses the maximum size and age of the entire population to normalize differences among individuals.

Observation:

Outputs include metabolism-rate, total-metabolism, and efficiency multipliers.

4.6 Initialization

Variable	Initialized Value	Justification
Met-base	Computed using size, age, and temperature factors	Establishes a physiologically realistic baseline.
swim-base	Met-base times movement ratio	Allocates baseline metabolism for locomotion.
E-base	Met-base times 0.02	Maintenance cost of cellular metabolic activity.
E-creation	Met-base times 0.05	Cost of forming new chloride cells.
motion-ratio	0.7 normalized	Movement is energetically dominant.
osmo-ratio	0.3 normalized	Osmoregulation is energetically significant.
E-osmo	0	No osmoregulation cost at initialization.

Variable	Initialized Value	Justification
metabolism-rate	Met-base	Baseline metabolism in the absence of environmental variation.
total-metabolism	0	Starts accumulation at zero.

4.7 Submodels

4.7.1 Initialize Metabolism

Baseline metabolic constant is computed using:

\$\$Met-base = base-constant (size / M-max) ^ 0.75 [1 + age-sensitivity * (1 - age / max-age)] * (optimal-temperature / max-survival-temp)\$\$

Where:

- base-constant is 0.1
- age-sensitivity is 0.5

4.7.2 Energy Partitions

Baseline energy partitions are computed from the initialized metabolic base:

$$\text{swim-base} = \text{Met-base} \cdot \text{motion-ratio}$$

$$E_{\text{base}} = \text{Met-base} \cdot 0.02$$

$$E_{\text{creation}} = \text{Met-base} \cdot 0.05$$

4.7.3 Metabolic Rate Calculation

At each tick, metabolism is recalculated using allometric, temperature, and age scaling:

$$\text{size-factor} = \left(\frac{\text{weight}}{\text{M-max}} \right)^{0.75}$$

$$\text{temp-factor} = Q_{10}^{\left(\frac{T_{\text{env}} - T_{\text{opt}}}{10}\right)}$$

$$\text{age-factor} = 1 + \left(1 - \frac{\text{age}}{\text{max-age}}\right) \cdot 0.5$$

Total metabolic rate:

$$M_{\text{total}} = \text{Met-base} \cdot \text{size-factor} \cdot \text{temp-factor} \cdot \text{age-factor}$$

Metabolic accumulation:

$$\text{total-metabolism} = \text{total-metabolism} + M_{\text{total}}$$

4.7.4 Efficiency Multipliers

A baseline metabolic reference is defined using:

- baseline size-factor = 1
- baseline age-factor = 1.5
- baseline temp-factor = 1

Thus:

$$M_{\text{baseline}} = \text{Met-base} \cdot 1 \cdot 1 \cdot 1.5$$

Efficiency is computed as:

$$\text{efficiency} = \frac{M_{\text{total}}}{M_{\text{baseline}}}$$

This value is applied to:

- **swim-efficiency**
- **osmoregulation-efficiency**
- **lipid-catabolism-efficiency**

Efficiency dynamically scales the cost of movement, ion regulation, and lipid catabolism.

4.8 Netlogo Implementation

```
;; =====  
;;   METABOLISM SUBMODEL  
;; =====  
  
globals [  
  ;; global physiological constants  
  Q10  
]  
  
patches-own [  
  temperature  
]  
  
fish-own [  
  ;; core state variables  
  weight  
  age-num  
  optimal-temperature  
  max-survival-temp  
  
  ;; metabolic scalars  
  Met-base  
  metabolism-rate  
  total-metabolism  
  
  ;; efficiency multipliers  
  swim-efficiency  
  osmoregulation-efficiency  
  lipid-catabolism-efficiency  
  
  ;; energy partitioning  
  swim-base  
  E-base  
  E-creation  
  E-osmo  
  motion-ratio  
  osmo-ratio  
  
  ;; population normalization  
  M-max  
  max-age  
]
```

```

;; =====
;;      INITIALIZE METABOLISM
;; =====

to initialize-metabolism
  let beta 0.75
  let age-sensitivity 0.5
  let base-constant 0.1

  ;; normalize population scaling
  set M-max max [weight] of fish
  set max-age max [age-num] of fish
  if M-max <= 0 [ set M-max 1 ]
  if max-age <= 0 [ set max-age 1 ]

  ;; scaling factors
  let size-factor (weight / M-max) ^ beta
  let age-factor 1 + age-sensitivity * (1 - (age-num / max-age))
  let temp-factor (optimal-temperature / max-survival-temp)

  ;; baseline metabolic constant
  set Met-base base-constant * size-factor * age-factor * temp-factor

  ;; energy partitions
  let motion-ratio-local 0.7
  let osmo-ratio-local 0.3
  let total-ratio (motion-ratio-local + osmo-ratio-local)

  set motion-ratio (motion-ratio-local / total-ratio)
  set osmo-ratio (osmo-ratio-local / total-ratio)

  set swim-base (Met-base * motion-ratio)
  set E-osmo 0

  ;; fixed fractions
  set E-base (Met-base * 0.02)
  set E-creation (Met-base * 0.05)

  ;; initial metabolic state
  set metabolism-rate Met-base
  set total-metabolism 0
end

;; =====

```

```

;;          CALCULATE METABOLISM EACH TICK
;; =====

to calculate-metabolism
  let beta 0.75

  ;; repopulate scaling values
  set M-max max [weight] of fish
  set max-age max [age-num] of fish
  if M-max <= 0 [ set M-max 1 ]
  if max-age <= 0 [ set max-age 1 ]

  ;; get patch temperature safely
  let T-env [temperature] of patch-here
  if not is-number? T-env [ set T-env optimal-temperature ]

  ;; scaling factors
  let size-factor (weight / M-max) ^ beta
  let temp-factor Q10 ^ ((T-env - optimal-temperature) / 10)
  let age-factor 1 + (1 - (age-num / max-age)) * 0.5

  ;; metabolic rate (per tick)
  let M-total Met-base * size-factor * temp-factor * age-factor
  set metabolism-rate M-total
  set total-metabolism total-metabolism + M-total

  ;; baseline reference metabolism
  let baseline-size-factor 1
  let baseline-age-factor 1.5
  let baseline-temp-factor 1

  let baseline-metabolism Met-base *
    baseline-size-factor *
    baseline-temp-factor *
    baseline-age-factor

  ;; efficiency multipliers
  let efficiency M-total / baseline-metabolism

  set swim-efficiency      efficiency
  set osmoregulation-efficiency efficiency
  set lipid-catabolism-efficiency efficiency
end

```


Chapter 5

Digestion

5.1 Overview

Digestion converts consumed prey mass into usable energy and transfers associated contaminants such as inorganic mercury and methylmercury from stomach contents into internal body burden pools. This submodel simulates the breakdown of ingested prey at a rate proportional to the agent's current metabolic rate. As stomach contents are digested, a fraction is converted into energy while mercury and methylmercury are incorporated into the agent's contaminant loads. When stomach contents reach zero, agents transition into foraging mode and begin drawing on lipid reserves.

5.2 Purpose

To simulate how migratory fish process ingested prey into energy and contaminants while linking digestive performance to metabolic rate, foraging behavior, and exposure risk. Digestion provides the energetic foundation required for movement, migration, and physiological processes while directly influencing contaminant accumulation.

5.3 Entities, State Variables, and Scales

5.3.1 Spatial and Temporal Scales

Spatial Unit: Patch (3 m x 3 m resolution)

Temporal Unit: 5 minute time steps (tick)

5.3.2 Global Variables

Global Variable	Definition
none required	Digestion uses internal agent state + metabolism-rate from metabolism submodel.

5.3.3 Patch Variables

Variable Name	Definition
None required	Digestion is an internal physiological process and does not depend on patch attributes.

5.3.4 Agent Variables

Variable Name	Definition
stomach-contents	Total mass of prey currently held in the stomach.
stomach-contents-Hg	Inorganic mercury contained within the stomach contents.
stomach-contents-MeHg	Methylmercury contained within the stomach contents.
digestion-rate	Rate of digestion equal to the current metabolic rate.
digestion-efficiency	Fraction of digested mass converted into usable energy.
energy	Total energetic reserves available to the agent.
gained-energy	Energy gained during the current tick from digestion.
mehg-foraging	Methylmercury released from stomach contents and absorbed during digestion.
hg-foraging	Inorganic mercury released from stomach contents during digestion.
mehg-foraging-total	Cumulative methylmercury risk through feeding.

Variable Name	Definition
hg-foraging-total	Cumulative inorganic mercury risk via digestion.
mehg-total	Total body burden of methylmercury risk.
hg-total	Total body burden of inorganic mercury risk.
foraging	Boolean indicating whether the agent is actively searching for prey.
lipid-loss	Boolean indicating when lipid reserves are used due to lack of stomach content.

5.4 Process Overview and Scheduling

1. Set digestion rate equal to the current metabolic rate.
2. Calculate the mass digested during the current tick.
3. Compute the corresponding amounts of inorganic and methylmercury digested.
4. Remove digested prey mass and associated contaminants from stomach contents.
5. Convert a fraction of digested biomass into usable energy.
6. Add mercury and methylmercury to cumulative body burden pools.
7. When stomach contents reach zero, set foraging to true and begin lipid use.

5.5 Design Concepts

Basic Principles:

Digestion is modeled as a metabolic rate dependent process, where higher metabolic demand results in greater digestive throughput. Contaminants are transferred into the agent's body burden proportionally to the fraction of mass digested.

Emergence:

Patterns of energy gain and contaminant uptake emerge from interactions among foraging success, metabolism, digestion efficiency, and prey availability.

Adaptation:

Agents respond to depleted stomach contents by transitioning into foraging mode and initiating lipid catabolism when necessary.

Objectives:

Agents maximize energetic intake to support migration, physiological function, and survival while minimizing energetic deficits.

Sensing:

Agents sense their own energetic state through stomach contents and energy levels.

Stochasticity:

Stochasticity arises indirectly from variation in prey encounters or foraging success, although digestion itself is deterministic.

Collectives:

Digestion occurs independently for each agent and does not require group level dynamics.

Observation:

Outputs of interest include digested mass per tick, gained energy, contaminant uptake, and cumulative contaminant burdens.

5.6 Initialization

Variable	Initialized Value	Justification
stomach-contents	0	Agents begin without prey unless defined by scenario.
stomach-contents-Hg	0	No initial inorganic mercury from digestion.
stomach-contents-MeHg	0	No initial methylmercury from digestion.
digestion-efficiency	Species specific	Determines fraction of biomass converted to energy.
foraging	true	Agents begin in search mode when stomach is empty.

5.7 Submodels

5.7.1 Digestion Rate and Breakdown

Digestive throughput is driven by metabolic rate:

$$\textit{digestion-rate} = \textit{metabolism-rate}$$

Digested mass per tick:

$$\textit{digested} = \textit{stomach-contents} \cdot \textit{digestion-rate}$$

Proportional mercury assimilation:

$$\textit{digested-Hg} = \left(\frac{\textit{digested}}{\textit{stomach-contents}} \right) \cdot \textit{stomach-contents-Hg}$$

$$\textit{digested-MeHg} = \left(\frac{\textit{digested}}{\textit{stomach-contents}} \right) \cdot \textit{stomach-contents-MeHg}$$

Updated stomach contents:

$$\textit{stomach-contents} = \textit{stomach-contents} - \textit{digested}$$

5.7.2 Energy Conversion

Energy gained from digestion:

$$\textit{gained-energy} = \textit{digested} \cdot \textit{digestion-efficiency} \cdot 40$$

Updated energy pool:

$$\textit{energy} = \textit{energy} + \textit{gained-energy}$$

5.7.3 Contaminant Assimilation

Contaminants absorbed during digestion:

$$\textit{mehg-foraging} = \textit{digested-MeHg}$$

$$hg\text{-foraging} = digested\text{-Hg}$$

Accumulated body burden:

$$mehg\text{-total} = mehg\text{-total} + digested\text{-MeHg}$$

$$hg\text{-total} = hg\text{-total} + digested\text{-Hg}$$

5.7.4 Transition to Foraging

When stomach contents reach zero:

- lipid-loss is activated

This switches the prey to burn fat energy reserves instead of relying on stomach contents.

5.8 Netlogo Implementation

```
;; =====
;; DIGESTION SUBMODEL
;; =====

fish-own [
  ;; digestion state
  stomach-contents
  stomach-contents-Hg
  stomach-contents-MeHg
  digestion-rate
  digestion-efficiency

  ;; contaminant tracking
  mehg-foraging
  hg-foraging
  mehg-foraging-total
  hg-foraging-total
  mehg-total
  hg-total
]
```

```

;; energy + biomass
energy
weight
lipid-loss
lipid-catabolism-efficiency
]

to digest
;; digestion tied to metabolic rate
set digestion-rate metabolism-rate

;; nothing to digest
if stomach-contents <= 0 [
  set foraging true
  set lipid-loss true
  stop
]

;; =====
;; DIGESTIVE BREAKDOWN
;; =====

;; amount of prey broken down this tick
let digested (stomach-contents * digestion-rate)

;; numerical guard to avoid divide-by-zero
if stomach-contents <= 0.000001 [ set digested 0 ]

let digested-Hg 0
let digested-MeHg 0

if digested > 0 and stomach-contents > 0 [
  set digested-Hg ((digested / stomach-contents) * stomach-contents-Hg)
  set digested-MeHg ((digested / stomach-contents) * stomach-contents-MeHg)
]

;; =====
;; UPDATE STOMACH CONTENTS
;; =====

set stomach-contents (max list 0 (stomach-contents - digested))
set stomach-contents-Hg (max list 0 (stomach-contents-Hg - digested-Hg))
set stomach-contents-MeHg (max list 0 (stomach-contents-MeHg - digested-MeHg))

;; =====

```

```

;; ENERGY CONVERSION
;; =====

let usable-energy (digested * digestion-efficiency * 40)
set energy energy + usable-energy

;; =====
;; STORE LIPID IF ENERGY SURPLUS
;; =====

if energy > 100 [
  let surplus (energy - 100)

  ;; convert surplus to lipid
  let lipid-added (surplus / 40)

  ;; apply storage efficiency
  set lipid-added (lipid-added * lipid-catabolism-efficiency)

  ;; update biomass
  set weight weight + lipid-added

  ;; subtract energy equivalent
  set energy energy - (lipid-added * 40)
]

;; =====
;; CONTAMINANT ASSIMILATION
;; =====

set mehg-foraging      digested-MeHg
set hg-foraging        digested-Hg
set mehg-foraging-total mehg-foraging-total + digested-MeHg
set hg-foraging-total  hg-foraging-total + digested-Hg
set mehg-total         mehg-total + digested-MeHg
set hg-total           hg-total + digested-Hg

;; =====
;; SWITCH TO Lipid Loss IF EMPTY
;; =====

if stomach-contents <= 0 [
  set lipid-loss true
]
end

```

Chapter 6

Salinity Exposure

6.1 Overview

Osmoregulation allows migratory fish to maintain homeostasis by regulating internal ion concentrations in response to varying environmental salinities. This function simulates salinity exposure through ion-regulatory stress, chloride cell expression, and the metabolic energy cost of to acclimate to this exposure internally in a spatiotemporal explicit context.

6.2 Purpose

To simulate stress response to salinity changes for migratory fish in coastal systems by regulating chloride cell density and allocating energy toward ion-regulatory processes.

6.3 Entities, State Variables, and Scales

6.3.1 Spatial and Temporal Scales

- **Spatial Unit:** Patch (3 m x 3 m resolution)
- **Temporal Unit:** 5-minute time steps (`tick`)

6.3.2 Global Variables

Global Variable	Definition
none required	Salinity exposure uses internal agent state

6.3.3 Patch Variables

Variable Name	Definition
Salinity S_{patch}	The salt concentration of a given patch, derived from hydrodynamic model inputs.

6.3.4 Agent Variables

Variable Name	Definition
acclimated-salinity S_{agent}	The salinity level the agent is currently acclimated to.
ionregulatory-stress I_{stress}	The level of stress an agent experiences when regulating ion balance due to osmotic difference.
chloride-density-min C_{min}	Minimum level of chloride cells, present even in low-stress conditions.
chloride-density-max C_{max}	Maximum level of chloride cells at high stress.
chloride-cell-density C	The current number of chloride cells expressed by the agent.
chloride-max-proliferation $R_{proliferation}$	The max number of chloride cells that can be expressed per time step.
chloride-cells-this-tick C_{tick}	The number of chloride cells created (or destroyed) in the current time step.
acclimation-rate α	The rate at which chloride cell density increases over time.
C-mid C_{mid}	The chloride cell density at which stress buffering is 50% effective.
time-since-last-osmoregulation t_{osmo}	The time elapsed since the last chloride cell regulation event.
Energy E_{agent}	The agent's total available energy for physiological functions.
E-osmo E_{osmo}	Total energy used for ion regulation (osmoregulation).
E-base E_{base}	The base energy cost per chloride cell.
E-creation $E_{creation}$	The energy cost for producing new chloride cells.

Variable Name	Definition
metabolic-max Met_{max}	Maximum metabolic cost for chloride cell creation.

6.4 Process Overview and Scheduling

1. Compute osmotic stress based on difference between S_{patch} and S_{agent} .
2. Adjust chloride cell density depending on time since last osmoregulation.
3. Compute energy cost of osmoregulation.
4. Deduct energy expenditure from agent's energy pool.

6.5 Design Concepts

Basic Principles: The model is based on principles of physiological ecology and osmoregulatory energetics in teleost and apterygian species. It draws from empirical findings (e.g., Allen et al., 2009; Little et al., 2023) and includes size scaling, stress buffering, and energy constraints. These principles are implemented at the submodel level (e.g., chloride proliferation, stress calculation) to simulate realistic physiological feedbacks to changes in environmental salinity.

Emergence: Ion-regulatory stress, chloride cell expression, and energy expenditure emerge from an agent's interaction with temporally and spatially variable salinity environments. These patterns are not pre-specified but arise dynamically through adaptive physiological responses.

Adaptation: Agents respond to osmotic stress by adjusting chloride cell density, a trait that buffers stress. This process allows individuals to reduce internal-external salinity gradients and maintain ion homeostasis.

Objectives: Agents seek to support survival by reducing stress and avoiding excessive energy loss through regulating chloride cell expression.

Sensing: Agents sense local salinity (S_{patch}) and compare it with their acclimated salinity (S_{agent}). They also track their own energy state and time since last osmoregulation.

Stochasticity: Acclimation may vary with α , which can be drawn from a defined range per individual to reflect physiological variation across the population.

Observation: Outputs include I_{stress} , C , E_{osmo} , and E_{agent} , all tracked per individual and exportable for analysis or visualization.

6.6 Initialization

Variable	Initialized Value	Justification
S_{patch}	user-defined for data input	This input can be user-defined realistic data values or known spatial data.
S_{agent}	35 (psu)	Assumes agents start acclimated to marine environment.
I_{stress}	1	Acclimated agents have minimal stress levels.
C	50%	Starts with partial cell density, allowing for regulation depending on environmental conditions.
C_{min}	10%	A baseline level of chloride cells is necessary for basic osmoregulatory functions.
C_{max}	100%	Agents can't express more than 100% of cells.
α	0.0017 - 0.002	Osmolarity stabilization from Figure 3. in (Allen et al., 2009).
C_{mid}	50%	When cells are 50% density, stress buffering is 50% effective (Allen et al., 2009).
E_{agent}	100%	Agent starts with limited energy before migration.
E_{base}	Teleost (4%) Aptoerygian ()	Based on the branchial cost (Little et al., 2023; Kirschner, 1993).
Met_{max}	Teleost (3.5%)	Based on the intestinal and renal cost & size of agent (Little et al., 2023; Kirschner, 1993).
k	-0.75	Scaling component for body mass is negative (Kirschner, 1993) and follows Kleiber's Law.

6.7 Submodels

6.7.1 Osmotic Stress

Ion-regulatory stress (I_{stress}) is calculated based on the difference between an agent's acclimated salinity and the ambient patch salinity, adjusted by the chloride cell buffering effect:

$$I_{stress} = \frac{\log_{10}(1 + |S_{agent} - S_{patch}|) \cdot 10}{1 + e^{-2 \cdot (C/C_{mid})}}$$

Stress is capped within the range $[1, 10]$, and may be reduced slightly over time if salinity remains stable and chloride density is sufficient:

$$I_{stress} = I_{stress} \cdot 0.98 \quad \text{if conditions are stable and } C > C_{min}$$

Agents also slowly shift their acclimated salinity toward ambient salinity when conditions have been stable for several time steps:

$$S_{agent} = S_{agent} + (S_{patch} - S_{agent}) \cdot 0.02$$

Where:

- I_{stress} is ion-regulatory (osmotic) stress, scaled between 1 and 10.
- S_{agent} is the agent's acclimated salinity.
- S_{patch} is the environmental salinity at the current patch.
- C is the chloride cell density (percent of maximum).
- C_{mid} is the density at which buffering is 50% effective.

6.7.2 Chloride Cell Density

Chloride cell proliferation is driven by the level of ion-regulatory stress the agent experiences when encountering a difference in salinity. The greater the stress, the higher the target chloride density the agent attempts to reach, up to a maximum threshold. Agents adjust their chloride cell density based on their current ion-regulatory stress and acclimation status. Chloride cells are not adjusted unless the agent's energy exceeds 25%.

The chloride cell density is based on stress:

$$C_{target} = C_{min} + (C_{max} - C_{min}) \cdot \left(\frac{I_{stress}}{10} \right)$$

If salinity conditions have remained stable for an extended period (e.g., 288 ticks), (C_{target}) is slightly reduced to reflect partial downregulation of chloride cells due to long-term acclimation:

$$C_{target} = C_{target} \cdot 0.99 \quad \text{if stable}$$

Chloride cell density then approaches the target using a double-rate adjustment and capped maximum rate of change:

$$\Delta C = (C_{target} - C_{current}) \cdot (2 \cdot R_{proliferation})$$

$$\Delta C = \max(-R_{max}, \min(R_{max}, \Delta C))$$

If the agent has low energy ($\leq 50\%$), the adjustment rate is halved:

$$\Delta C = \Delta C \cdot 0.5 \quad \text{if energy is low}$$

Finally, the chloride cell density is updated and constrained between C_{min} and C_{max} :

$$C_{new} = \max(C_{min}, \min(C_{max}, C_{current} + \Delta C))$$

This ensures that the agent does not overshoot the physiologically realistic limit of chloride cell density, while still responding to osmotic stress.

Chloride density is only recalculated after a given acclimation interval:

$$t_{osmo} \geq \alpha^{-1}$$

After updating, the acclimation timer is reset:

$$t_{osmo} = 0$$

This prevents agents from recalculating chloride density every time step and allows for controlled, realistic responses to prolonged stress and salinity changes.

Where:

- I_{stress} is the ion-regulatory stress, scaled from 1 to 10.
- C_{target} is the desired chloride cell density based on stress level.
- C_{min} and C_{max} are the bounds for chloride cell density.
- $R_{proliferation}$ determines the **maximum allowable increase** per time step.
- ΔC is the rate of change in chloride cell expression.
- $R_{max} = (C_{max} - C_{min}) \cdot R_{proliferation}$
- C_{new} is the percent of new chloride cell expression.
- α is the acclimation rate constant.
- t_{osmo} represents time since the last osmoregulation event.

Osmoregulation Energy

Metabolic cost related to size:

$$E_{creation} = Met_{max} * (\frac{M}{M_{max}})^k$$

Where:

- $E_{creation}$ is the energy cost of chloride cell creation
- Met_{max} is the maximum metabolic cost of the agent

- M is equal to the agent's size, where smaller fish spend proportionally more energy on osmoregulation (Little et al., 2023)
- M_{max} is the maximum mass of an agent within the population
- k follows size-dependent variation in energy allocation, consistent with a negative scaling exponent.

Energy required for ion regulation:

$$E_{osmo} = (E_{base} \cdot C) + (E_{creation} \cdot C_{tick})$$

Where:

- E_{base} represents the energy cost per chloride cell for maintenance.
- C is the current chloride density.
- $E_{creation}$ represents the cost of producing new chloride cells.
- C_{tick} is the number of newly created chloride cells in the current time step.

6.7.3 Energy Balance

Agents balance energy to osmoregulate with total energy allowance:

$$E_{agent} = E_{agent} - E_{osmo}$$

Where:

- E_{agent} is the total energy of the agent.
- E_{osmo} is the energy consumed during osmoregulation.

```
;; =====
;; SALINITY EXPOSURE AND OSMOREGULATION SUBMODEL
;; =====

globals [
    stable-time           ;; tracks duration of stable salinity
    salinity-threshold    ;; threshold difference to count as "stable"
]

patches-own [
    salinity              ;; environmental salinity (psu)
```

```

]

fish-own [
  ;; --- Salinity Physiology ---
  acclimated-salinity
  ionregulatory-stress      ;; 1-10 scale

  ;; --- Chloride Cell Physiology ---
  chloride-density-min
  chloride-density-max
  chloride-cell-density
  chloride-max-proliferation
  chloride-cells-this-tick
  C-mid

  ;; --- Acclimation Timing ---
  acclimation-rate
  time-since-last-osmoregulation

  ;; --- Energetics ---
  energy
  E-osmo
  E-base
  E-creation
  Met-max

  ;; --- Size Scaling ---
  weight
]

;; =====
;; MAIN OSMOREGULATION PROCEDURE
;; =====

to osmoregulate
  calculate-ionregulatory-stress
  regulate-chloride-cell-density
  calculate-osmoregulation-energy
end

;; =====
;; STEP 1 - CALCULATE ION-REGULATORY STRESS
;; =====

```

```

to calculate-ionregulatory-stress
  let prev-sal [salinity] of previous-patch
  let patch-sal [salinity] of patch-here
  let agent-sal acclimated-salinity

  ;; --- Base salinity difference ---
  let sal-diff abs (agent-sal - patch-sal)

  ;; --- Core salinity-based stress function ---
  let salinity-stress salinity-to-stress sal-diff chloride-cell-density ionregulatory-stress

  ;; --- Salinity stability, used for acclimation ---
  let sal-change abs (patch-sal - prev-sal)

  if sal-change < salinity-threshold [
    set stable-time stable-time + 1
  ]
  if sal-change >= salinity-threshold [
    set stable-time 0
  ]

  ;; --- Slow acclimation when salinity is stable ---
  if stable-time > 4 and chloride-cell-density > chloride-density-min [
    set salinity-stress salinity-stress * 0.98
    set acclimated-salinity acclimated-salinity +
      ((patch-sal - acclimated-salinity) * 0.02)
  ]

  ;; --- Chloride buffering ---
  let buffer-reduct calculate-stress-reduction-chloride-function chloride-cell-density

  ;; Buffering only applies above minimum chloride levels
  if chloride-cell-density > chloride-density-min [
    set ionregulatory-stress max (list 1 (salinity-stress - buffer-reduct * 0.8))
  ]

  if chloride-cell-density <= chloride-density-min [
    set ionregulatory-stress salinity-stress
  ]
end

;; =====
;; STEP 2 - REGULATE CHLORIDE CELL DENSITY
;; =====

```

```

to regulate-chloride-cell-density
  ;; energetic constraint
  if energy <= 25 [ stop ]

  set time-since-last-osmoregulation time-since-last-osmoregulation + 1

  ;; wait until acclimation interval
  if time-since-last-osmoregulation < acclimation-rate [ stop ]

  ;; --- Target chloride cell density ---
  let target-dens chloride-density-min +
    ((chloride-density-max - chloride-density-min) *
     (ionregulatory-stress / 10))

  ;; down-regulate if long-term stability
  if stable-time > 288 [
    set target-dens (target-dens * 0.99)
  ]

  ;; constrain target to bounds
  set target-dens max (list chloride-density-min
                        min (list chloride-density-max target-dens))

  ;; --- Adjustment rate ---
  let change (target-dens - chloride-cell-density) *
    (2 * chloride-max-proliferation)

  let max-change ((chloride-density-max - chloride-density-min) *
                  chloride-max-proliferation)

  ;; clamp change
  set change max (list (-1 * max-change) (min (list max-change change)))

  ;; energy-dependent reduction
  if energy <= 50 [
    set change (change * 0.5)
  ]

  ;; apply change
  set chloride-cell-density chloride-cell-density + change
  set chloride-cell-density max (list chloride-density-min
                                      min (list chloride-density-max chloride-cell-density))

  set chloride-cells-this-tick abs change
  set time-since-last-osmoregulation 0

```

```

end

;; =====
;; STEP 3 - OSMOREGULATION ENERGY COST
;; =====

to calculate-osmoregulation-energy
  let maintenance (E-base * chloride-cell-density)
  let creation     (E-creation * chloride-cells-this-tick)

  set E-osmo (maintenance + creation)
  set energy max (list 0 (energy - E-osmo))
end

;; =====
;; CHLORIDE BUFFERING FUNCTION
;; =====

to-report calculate-stress-reduction-chloride-function [C]
  ;; logistic-like buffering curve
  let reduction 5 * (1 - exp(C * -0.5 / C-mid))
  report reduction
end

;; =====
;; SALINITY → STRESS FUNCTION
;; =====

to-report salinity-to-stress [diff C prev-stress]
  let scaled (log (1 + abs diff) 10) * 10
  let buffer  (1 + exp(-2 * (C / C-mid)))
  let result  (scaled / buffer)

  ;; constrain to 1-10
  set result max (list 1 (min (list 10 result)))

  ;; acclimation effect
  if stable-time > 288 [
    set result (result * 0.99)
  ]

  report result

```

```
end

;; =====
;; SIZE-SCALED CREATION COST
;; =====

to calculate-E-creation
  let M-max max [weight] of fish
  if M-max <= 0 [ set M-max weight ]

  let beta -0.75
  set E-creation Met-max * ((weight / M-max) ^ beta)
end
```


Chapter 7

Migration Cues

7.1 Overview

This submodel simulates how migratory fish detect environmental cues that signal the onset of migration. The mechanism is grounded in empirical evidence that **simultaneous declines in daylength and temperature** reliably precede the initiation of seaward migration in several anadromous species. In this model, migration begins only when a fish experiences environmental conditions consistent with an autumnal cue window defined by:

- A decline in daylength

$$\Delta L < 0$$

- A decline in temperature

$$\Delta T < 0$$

- And these declines occurring **on the same day**, indicating a **coincident decline**

When this occurs, the day is labeled a *cue day*. Upon reaching cue days, individuals begin stochastically accumulating the probability of initiating migration. For predators (striped bass), migration probability additionally depends on alewife presence, linking species interactions to migration onset.

7.2 Purpose

The purpose of this submodel is to mechanistically trigger migration using **real, observed environmental phenology** rather than arbitrary or date-based rules. This framework captures how:

- **Changes in Δ Temperature and Δ Daylength structure seasonality**
- **Migration timing emerges pragmatically rather than being fixed**
- **Fish respond to external conditions and species interactions**
- **Environmental preprocessing in R feeds into NetLogo behavioral rules**

This allows migration in the model to originate from measurable ecological drivers instead of hard-coded thresholds.

7.3 Entities, State Variables, and Scales

7.3.1 Spatial and Temporal Scales

- **Spatial Unit:** Patch (3 m x 3 m resolution)
- **Temporal Unit:** 5-minute time steps (`tick`)

7.3.2 Global Variables

Global Variable	Definition
cue-table	Lookup table containing Δ Temperature, Δ Daylength, their smoothed values, and Boolean coincident-decline flags (from R preprocessing)
cue-days	List of day-of-year values where $\Delta T < 0$ and $\Delta L < 0$ simultaneously (identified in R)
ΔT	Daily rate of change in temperature imported from R (<code>dT_per_day_f</code> or smoothed <code>dT_smooth</code>)

Global Variable	Definition
ΔL	Daily rate of change in photoperiod imported from R (dL_per_day_min or smoothed dL_smooth)
cue-active?	Whether the current day-of-year corresponds to a coincident decline day
migration-trigger?	Whether environmental signals are strong enough to begin building migration probability

7.3.3 Patch Variables

Migration cues do not depend on patch-level conditions. No patch-scale variables are required.

7.3.4 Agent Variables

Variable	Definition
migration-trigger?	Whether the agent has entered a window where migration probability begins to accumulate
migration-probability	The cumulative probability of initiating migration, updated each tick once the trigger is active
start-migration?	Whether the individual has begun active migration

7.4 Process Overview and Scheduling

7.4.1 1. Environmental Inputs (from R)

External R preprocessing computes:

- Δ Temperature per day
- Δ Daylength per day

- Smoothed 3-day rolling means
- Boolean `both_negative` indicating coincident decline
- Exports full dataset as "`cue_dataset_full.csv`"

This dataset defines the environmental basis for triggering migration.

7.4.2 2. Identify Cue Days

Cue days are those where:

$$\Delta T < 0 \quad \wedge \quad \Delta L < 0$$

The model loads the CSV at setup and extracts all days meeting this criterion into `cue-days`.

7.4.3 3. Activate Cue

Each simulation day checks whether:

$$\text{day-of-year} \in \text{cue-days}$$

If true, then:

- `cue-active? = true`
- `migration-trigger? = true`

This marks the environmental onset of migration preparedness.

7.4.4 4. Alewife Migration Probability Accumulation

Once the migration trigger is active, each tick adds:

$$\text{migration_probability}_{t+1} = \text{migration_probability}_t + (0.00001 + U(0, 0.00004))$$

Capped at:

$$\text{migration_probability} \leq 1$$

This produces:

- Individual variation in migration timing
- Realistic ramping into migration
- Stochastic onset probabilities

7.4.5 5. Initiation of Migration

An agent begins migration when:

$$\text{random-float}(1) < \text{migration_probability}$$

At this moment:

- **start-migration? = true**
- Movement functions shift from staging to directed migration

7.5 Design Concepts

Basic Principles

The mechanism reflects literature on photoperiod- and temperature-driven migration cues.

Emergence

Migration timing varies among individuals because probability increments differ slightly with stochasticity.

Observation

Key outputs: cue day activation, probability trajectories, proportion of migrants through time.

7.6 Initialization

Variable	Initial Value	Justification
cue-days	Extracted from CSV	Based on empirical environmental data
migration-probability	0	Ensures probability is accumulated only within cue window
migration-trigger?	false	No cue until ΔT and ΔL decline

Variable	Initial Value	Justification
cue-active?	false	Depends on DOY
start-migration?	false	Fish begin in staging mode

7.7 Submodels

7.8 1. Identifying Cue Days (Preprocessed in R)

$$\Delta T < 0, \Delta L < 0$$

The preprocessing script:

1. Computes ΔT and ΔL
2. Applies smoothing
3. Flags coincident declines

7.9 2. Alewife Migration Probability

$$migration_probability_{t+1} = migration_probability_t + (0.00001 + U(0, 0.00004))$$

$$migration_probability \leq 1$$

7.10 Netlogo Implementation

```
;; -----
;; GLOBALS
;; -----
globals [
  cue-table
  cue-days

  migration-trigger?
  cue-active?
]
;; -----
```

```

;; TURTLE VARIABLES
;; -----
turtles-own [
  migration-probability
  start-migration?
]

;; -----
;; LOAD CUE DATA (CSV FROM R)
;; -----
to load-cue-csv
  set cue-table table:make
  let raw csv:from-file "inputs/cue_dataset_full.csv"

  let header first raw
  let rows but-first raw

  foreach rows [
    r ->
      let doy      item 7 r
      let dTsm     item 21 r
      let dLsm     item 22 r
      let negflag  item 24 r

      table:put cue-table doy (list dTsm dLsm negflag)
  ]
end

;; -----
;; IDENTIFY CUE DAYS ( $\Delta T < 0$  AND  $\Delta L < 0$ )
;; -----
to find-migration-days
  set cue-days []

  foreach sort table:keys cue-table [
    d ->
      let vals table:get cue-table d
      let negflag item 2 vals

      if (negflag = "TRUE" or negflag = true) [
        set cue-days lput d cue-days
      ]
  ]
end

```

```

;; -----
;; UPDATE MIGRATION CUE (DAILY)
;; -----
to update-migration-cue
  ifelse member? day cue-days [
    set cue-active? true
    set migration-trigger? true
  ] [
    set cue-active? false
  ]
end

;; -----
;; ALEWIFE MIGRATION PROBABILITY ACCUMULATION
;; -----
to calc-migration-probability
  if migration-trigger? [
    set migration-probability
      migration-probability
      + (0.00001 + random-float 0.00004)

    if migration-probability > 1 [
      set migration-probability 1
    ]
  ]
end

to migrate
  if random-float 1 <= migration-probability
  set start-migration? true
end

```


Chapter 8

Contaminant Exposure

8.1 Overview

The contaminant exposure submodel simulates mercury and methylmercury exposure risk across gills for migratory fish moving through spatially and temporally variable contamination fields. Exposure risk depends on contaminant concentration, suspended particulate matter, and metabolic rate. Concentrations of mercury and methylmercury are normalized across the spatial domain, and depth averaged SPM values modulate risk to reflect particle bound contaminant transport. Agents accumulate both instantaneous and cumulative exposure, allowing estimation of bioaccumulation risk over the migration season.

8.2 Purpose

To evaluate mercury and methylmercury exposure and bioaccumulation risk for migratory fishes by integrating patch level contamination, ionregulatory stress, depth averaged suspended particulate matter, and metabolism driven uptake processes in estuarine and coastal systems.

8.3 Entities, State Variables, and Scales

8.3.1 Spatial and Temporal Scales

Spatial Unit: Patch (3 m x 3 m resolution)

Temporal Unit: 5 minute time steps (tick)

8.3.2 Global Variables

Variable	Initialized Value	Justification
MeHg-threshold	15 ug per kg	Screening threshold for methylmercury [Administration, Gaudet et al.]
Hg-threshold	150 ug per kg	NOAA guideline for mercury contamination [Administration, Gaudet et al.]
min-Hg, max-Hg	Calculated from input layers	Required for Hg normalization across domain
min-MeHg, max-MeHg	Calculated from input layers	Required for MeHg normalization across domain

8.3.3 Patch Variables

Variable Name	Definition
mercury	Mercury concentration at this patch.
methylmercury	Methylmercury concentration at this patch.
SPM	Depth averaged suspended particulate matter at this patch.
Hg-exp-risk-alewife	Patch level mercury exposure risk from alewives.
Hg-exp-risk-stripedbass	Patch level mercury exposure risk from striped bass.
MeHg-exp-risk-alewife	Patch level methylmercury risk from alewives.
MeHg-exp-risk-stripedbass	Patch level methylmercury risk from striped bass.

8.3.4 Agent Variables

Variable Name	Definition
metabolism-rate	Current metabolic rate used to scale uptake risk.
hg-exposure-duration	Number of ticks spent above the mercury threshold.
mehg-exposure-duration	Number of ticks spent above the MeHg threshold.
hg-uptake-risk	Instantaneous mercury uptake risk per tick.
mehg-uptake-risk	Instantaneous methylmercury uptake risk per tick.
hg-total	Cumulative mercury uptake.
mehg-total	Cumulative methylmercury uptake.
hg-exposure-total	Total non normalized Hg exposure.
mehg-exposure-total	Total non normalized MeHg exposure.
hg-exposure-total-normalized	Sum of normalized Hg exposure values.
mehg-exposure-total-normalized	Sum of normalized MeHg exposure values.

8.4 Process Overview and Scheduling

1. Retrieve mercury, methylmercury, and depth averaged SPM values from the current patch.
2. Normalize patch contaminants between 0 and 1.
3. Compare patch levels to stress thresholds.
4. Update exposure durations.
5. Compute uptake risk using metabolism, ionregulatory stress, and SPM.
6. Add uptake risk to cumulative body burden.
7. Update patch level species specific risk metrics.

8.5 Design Concepts

Basic Principles:

Contaminant uptake follows toxicokinetic principles where absorption increases with concentration, metabolic rate, and suspended particulate matter (SPM). Depth averaged SPM reflects hydrodynamic transport and particle bound contaminant processes for fish typically found near the middle of the water column.

Emergence:

Exposure and risk patterns emerge from individual movement across heterogeneous contamination fields coupled with metabolic and stress driven physiology.

Adaptation:

Agents do not actively avoid contamination, but their internal physiology modulates uptake in response to metabolic changes.

Objectives:

This model does not give agents risk minimization behavior. Instead, risk is assessed to evaluate means of exposure pathways and cumulative toxic exposure.

Sensing:

Agents sense patch level mercury, methylmercury, SPM, and internal metabolic state.

Stochasticity:

Variation in spatial contaminant fields or individual characteristics and state creates stochastic variation in exposure.

Observation:

Outputs include instantaneous exposure risk, cumulative exposure risk, normalized exposure totals, and patch level species specific contributions.

8.6 Initialization

Variable	Initialized Value	Justification
mercury	Input layer	Hydrodynamic or field derived contaminant layer.
methylmercury	Input layer	Same as above.
SPM	Depth averaged input	Reflects hydrodynamic particle transport.

8.7 Submodels: Mercury Exposure and Uptake

8.7.1 Normalization

$$Hg_{normalized} = \frac{Hg_{current} - minHg}{maxHg - minHg}$$

Bounded in the range 0 to 1.

8.7.2 Exposure Duration

$$Hg_{exp} = Hg_{exp} + 1 \quad \text{if } Hg_{current} > Hg_{threshold}$$

8.7.3 Uptake Risk

$$Hg_{risk} = metabolism \cdot Hg_{normalized} \cdot (1 + S) \cdot (1 + SPM)$$

8.7.4 Cumulative Uptake

$$Hg_{total} = Hg_{total} + Hg_{risk}$$

Patch level storage based on species for cumulative duration risk experience in space:

- Hg-exp-risk
- Hg-exp-risk

8.8 Submodels: Methylmercury Exposure and Uptake

8.8.1 Normalization

$$MeHg_{normalized} = \frac{MeHg_{current} - minMeHg}{maxMeHg - minMeHg}$$

Clamped to the range 0 to 1.

8.8.2 Exposure Duration

$$MeHg_{exp} = MeHg_{exp} + 1 \quad \text{if } MeHg_{current} > MeHg_{threshold}$$

8.8.3 Uptake Risk

$$MeHg_{risk} = metabolism \cdot MeHg_{normalized} \cdot (1 + S) \cdot (1 + SPM)$$

8.8.4 Cumulative Uptake

$$MeHg_{total} = MeHg_{total} + MeHg_{risk}$$

Patch level storage based on species for cumulative duration risk experience in space:

- MeHg-exp-risk
- MeHg-exp-risk

```
globals [
  ;; Thresholds
  MeHg-threshold
  Hg-threshold

  ;; Domain-wide normalization bounds
  min-Hg
  max-Hg
  min-MeHg
  max-MeHg
]

patches-own [
  ;; Contaminant inputs
  mercury
  methylmercury
  SPM

  ;; Species-specific instantaneous risk
  Hg-exp-risk-alewife
  Hg-exp-risk-stripedbass
  MeHg-exp-risk-alewife
  MeHg-exp-risk-stripedbass

  ;; Species-specific cumulative patch totals
  Hg-exp-risk-total-alewife
  Hg-exp-risk-total-stripedbass
  MeHg-exp-risk-total-alewife
  MeHg-exp-risk-total-stripedbass
```

```

]

turtles-own [
  ;; Physiology
  metabolism-rate
  ionregulatory-stress

  ;; Exposure durations
  hg-exposure-duration
  mehg-exposure-duration

  ;; Instantaneous uptake
  hg-uptake-risk
  mehg-uptake-risk

  ;; Cumulative uptake
  hg-total
  mehg-total

  ;; Cumulative exposure raw
  hg-exposure-total
  mehg-exposure-total

  ;; Cumulative normalized exposure
  hg-exposure-total-normalized
  mehg-exposure-total-normalized
]

;; =====
;; MERCURY EXPOSURE
;; =====

to mercury-contamination

  ;; -----
  ;; Retrieve patch contaminant and SPM values
  ;; -----
  let Hg-current [mercury] of patch-here
  let SPM-current [SPM] of patch-here

  if not is-number? Hg-current [ set Hg-current 0 ]
  if not is-number? SPM-current [ set SPM-current 0 ]

  ;; -----
  ;; Normalize Hg concentration (0-1)

```

```

;; -----
let Hg-normalized 0
if (max-Hg - min-Hg) > 0 [
  set Hg-normalized (Hg-current - min-Hg) / (max-Hg - min-Hg)
]

;; clamp
set Hg-normalized max list 0 (min list 1 Hg-normalized)

;; -----
;; Exposure duration: threshold exceedance
;; -----
if Hg-current > Hg-threshold [
  set hg-exposure-duration hg-exposure-duration + 1
]

;; -----
;; Uptake risk (toxicokinetic rule)
;; -----
let stress-factor (1 + ionregulatory-stress)
let spm-factor    (1 + SPM-current)

set hg-uptake-risk (metabolism-rate * Hg-normalized * stress-factor * spm-factor)

;; no negative risk ever
if hg-uptake-risk < 0 [ set hg-uptake-risk 0 ]

;; -----
;; Add to cumulative body burden
;; -----
set hg-total hg-total + hg-uptake-risk

;; -----
;; Track cumulative raw & normalized exposure
;; -----
set hg-exposure-total      hg-exposure-total + Hg-current
set hg-exposure-total-normalized hg-exposure-total-normalized + Hg-normalized

;; -----
;; Patch-level risk recording (species specific)
;; -----
let patch-risk hg-uptake-risk

if breed = alewives [
  ask patch-here [

```



```

    set Hg-exp-risk-alewife      patch-risk
    set Hg-exp-risk-total-alewife (Hg-exp-risk-total-alewife + patch-risk)
  ]
]

if breed = stripedbass [
  ask patch-here [
    set Hg-exp-risk-stripedbass      patch-risk
    set Hg-exp-risk-total-stripedbass (Hg-exp-risk-total-stripedbass + patch-risk)
  ]
]

end

;; =====
;; METHYLMERCURY EXPOSURE
;; =====

to methylmercury-contamination

  ;; -----
  ;; Retrieve patch contaminant and SPM values
  ;; -----
  let MeHg-current [methylmercury] of patch-here
  let SPM-current  [SPM] of patch-here

  if not is-number? MeHg-current [ set MeHg-current 0 ]
  if not is-number? SPM-current  [ set SPM-current 0 ]

  ;; -----
  ;; Normalize MeHg concentration (0-1)
  ;; -----
  let MeHg-normalized 0
  if (max-MeHg - min-MeHg) > 0 [
    set MeHg-normalized (MeHg-current - min-MeHg) / (max-MeHg - min-MeHg)
  ]

  ;; clamp
  set MeHg-normalized max list 0 (min list 1 MeHg-normalized)

  ;; -----
  ;; Exposure duration: threshold exceedance
  ;; -----
  if MeHg-current > MeHg-threshold [

```

```

    set mehg-exposure-duration mehg-exposure-duration + 1
  ]

  ;; -----
  ;; Uptake risk
  ;; -----
  let stress-factor (1 + ionregulatory-stress)
  let spm-factor    (1 + SPM-current)

  set mehg-uptake-risk (metabolism-rate * MeHg-normalized * stress-factor * spm-factor)

  if mehg-uptake-risk < 0 [ set mehg-uptake-risk 0 ]

  ;; -----
  ;; Add to cumulative body burden
  ;; -----
  set mehg-total mehg-total + mehg-uptake-risk

  ;; -----
  ;; Track cumulative raw & normalized exposure
  ;; -----
  set mehg-exposure-total      mehg-exposure-total + MeHg-current
  set mehg-exposure-total-normalized mehg-exposure-total-normalized + MeHg-normalized

  ;; -----
  ;; Patch-level risk recording (species specific)
  ;; -----
  let patch-risk mehg-uptake-risk

  if breed = alewives [
    ask patch-here [
      set MeHg-exp-risk-alewife      patch-risk
      set MeHg-exp-risk-total-alewife (MeHg-exp-risk-total-alewife + patch-risk)
    ]
  ]

  if breed = stripedbass [
    ask patch-here [
      set MeHg-exp-risk-stripedbass      patch-risk
      set MeHg-exp-risk-total-stripedbass (MeHg-exp-risk-total-stripedbass + patch-risk)
    ]
  ]

end

```

Chapter 9

Filter Feeding

9.1 Overview

Filter feeding is a foraging behavior in which fish consume suspended particulate matter as they migrate through estuarine and coastal systems. In this submodel, agents evaluate neighboring patches based on suspended particulate matter concentration and selectively filter feed where energetic return is highest. Filtration capacity increases with body size, metabolic rate, and digestion efficiency, and is reduced under competition from other filter feeding fish. Contaminant loads in the water column directly influence mercury and methylmercury assimilation during feeding, including biomagnification effects.

9.2 Purpose and Patterns

The purpose of this submodel is to simulate ecologically realistic filter feeding behavior in migratory fish using suspended particulate matter as a food resource. This behavior links hydrodynamic particle fields to energy gain, contaminant exposure, and movement decisions. Filter feeding only occurs when spatial conditions and energetic returns are favorable, and incorporates competition, biomagnification, and digestion constraints.

This submodel represents the following ecological patterns:

- **Filtration capacity increases with body size, metabolism, and digestive efficiency**
Larger individuals with higher metabolism filter more water and assimilate more prey mass.

- **Gaussian SPM preference centered on optimal SPM**
Feeding intensity is highest near an optimal SPM range and decreases away from that peak.
- **Crowding penalty from con-specifics**
Filtration benefit decreases when multiple individuals feed in the same patch.
- **Movement toward highest benefit**
Fish move toward the patch with the highest filtration benefit based on SPM and competitive pressure.
- **Contaminant assimilation tied directly to filtration**
Mercury and methylmercury are assimilated proportionally to SPM concentration and biomagnification factors derived from empirical trophic magnification relationships.

9.3 Entities, State Variables, and Scales

9.3.1 Spatial and Temporal Scales

Spatial Unit: Patch (3 m x 3 m resolution)

Temporal Unit: 5 minute time steps (tick)

9.3.2 Global Variables

Global Variable	Definition
spm-sd	Standard deviation for Gaussian SPM preference curve (controls width of feeding preference)
max-filter-distance	Maximum distance within which patches can be evaluated (optional, model dependent)
BMF-MeHg	Biomagnification factor for methylmercury (1.74; after Lavoie et al. 2013)
BMF-Hg	Biomagnification factor for inorganic Hg (1.38; after Lavoie et al. 2013)
previous-x	X-coordinate of agent's previous location (used to compute travel distance and energy cost)
previous-y	Y-coordinate of agent's previous location (used with previous-x)

9.3.3 Patch Variables

Variable Name	Definition
SPM	Suspended particulate matter concentration in the water column
mercury	Inorganic mercury concentration at the patch
methylmercury	Methylmercury concentration at the patch
patch-terrain	Identifies water or land patches

9.3.4 Agent Variables

Variable Name	Definition
size	Body size used to scale filtration capacity
weight	Biomass used in base filtration scaling (from metabolism submodel)
metabolism-rate	Metabolic rate that modulates filtration intensity
digestion-efficiency	Fraction of filtered biomass digested
optimal-SPM	SPM concentration where filtration is most efficient
filter-feed?	Boolean indicating whether the agent engages in filter feeding
energy	Current energy reserves
E-swim	Energy cost of swimming per meter
swim-efficiency	Efficiency of movement based on metabolic state
speed	Swimming speed
stomach-contents	Biomass currently held in the stomach
stomach-capacity	Maximum biomass that can be held
stomach-contents-Hg	Inorganic mercury within stomach contents
stomach-contents-MeHg	Methylmercury within stomach contents
hg-total	Total cumulative inorganic mercury body burden
mehg-total	Total cumulative methylmercury body burden

Variable Name	Definition
hg-exposure-total	Cumulative Hg exposure from all pathways
mehg-exposure-total	Cumulative MeHg exposure from all pathways
hg-exposure-total-normalized	Normalized inorganic mercury exposure
mehg-exposure-total-normalized	Normalized methylmercury exposure
previous-x (<i>if not global</i>)	Previous x-location for travel-distance calculation
previous-y (<i>if not global</i>)	Previous y-location

9.4 Process Overview and Scheduling

1. Compute filtration capacity based on size, metabolism, and digestion efficiency.
2. Evaluate neighboring patches using a Gaussian suitability curve around optimal SPM.
3. Apply a competition penalty based on the number of con-specifics feeding on each patch.
4. Select the patch with the highest filtration benefit.
5. Move toward the chosen patch with travel distance reduced by swimming efficiency.
6. Apply swimming energy cost.
7. Filter feed by assimilating SPM scaled biomass from the chosen patch.
8. Assimilate mercury and methylmercury using biomagnification factors from Lavoie et al (2013).
9. Add assimilated biomass and contaminants to stomach contents.

9.5 Design Concepts

Basic Principles

Filter feeding reflects how pelagic and estuarine fish forage on suspended par-

ticles while migrating. Filtration scales with size, metabolism, and digestive efficiency. SPM determines food availability and contaminant exposure.

Emergence

Feeding hotspots emerge where hydrodynamic SPM fields align with optimal SPM and competition is minimal. Contaminant loads emerge from local mercury and methylmercury concentrations and biomagnification.

Adaptation

Agents adjust their direction and feeding behavior based on SPM quality, filtration capacity, and local competition.

Sensing

Agents sense SPM, mercury, methylmercury, competition from conspecifics, stomach fullness, and their own metabolic and digestive state.

Stochasticity

Variation in SPM quality, contaminant concentrations, and local competition create variability in feeding and assimilation outcomes.

Interaction

Competition reduces filtration benefit when more fish feed on the same patch. Mercury and methylmercury assimilation depends on local contaminant levels and feeding intensity.

Observation

The model tracks biomass intake, mercury and methylmercury assimilation, stomach contents, and contaminant load added per patch.

9.6 Initialization

Variable	Initialized Value	Justification
stomach-contents	0	No biomass in the stomach at model start
stomach-contents-Hg	0	No initial inorganic mercury from feeding
stomach-contents-MeHg	0	No initial methylmercury
digestion-efficiency	Species specific	Determines fraction of filtered biomass assimilated
optimal-SPM	Species specific	Represents preferred SPM concentration

9.7 Submodels

9.7.1 Filtration Capacity

Filtration scales with size, metabolism, and digestion efficiency:

$$Filtration_{base} = \left(\frac{size}{100} \right) \cdot metabolism_rate \cdot digestion_efficiency$$

9.7.2 SPM Interest (Gaussian Suitability)

Suitability peaks at the optimal SPM and declines with distance:

$$Interest = e^{-\frac{(SPM - optimalSPM)^2}{2 \cdot \sigma^2}}$$

Where σ is the SPM standard deviation tolerance (here 25).

9.7.3 Benefit Calculation

Benefit for each neighboring patch:

$$Benefit = SPM \cdot Interest \cdot Filtration_{base}$$

If con-specifics are present:

$$Benefit = \frac{Benefit}{N_{conspecifics}}$$

9.7.4 Movement

Travel distance is reduced by swimming efficiency:

$$Distance = \min(DistanceToPatch, speed \cdot swim_efficiency)$$

Energy cost of swimming:

$$Energy = Energy - (E_swim \cdot Distance)$$

9.7.5 Feeding and Stomach Loading

Assimilated biomass:

$$Intake = \min(PreyAssimilated, StomachCapacity - StomachContents)$$

9.7.6 Contaminant Assimilation and Biomagnification

Biomagnification factors follow Lavoie et al.:

$$BMF_{MeHg} = 10^{(0.24 \cdot \Delta TP)}$$

$$BMF_{Hg} = 10^{(0.14 \cdot \Delta TP)}$$

With $\Delta TP = 1$:

$$BMF_{MeHg} = 10^{0.24} \approx 1.74$$

$$BMF_{Hg} = 10^{0.14} \approx 1.38$$

Assimilation equations:

$$MeHg_{assim} = SPM \cdot MeHg_{patch} \cdot BMF_{MeHg} \cdot PreyAssimilated$$

$$Hg_{assim} = SPM \cdot Hg_{patch} \cdot BMF_{Hg} \cdot PreyAssimilated$$

Stomach contaminant update:

$$StomachMeHg = StomachMeHg + MeHg_{assim}$$

$$StomachHg = StomachHg + Hg_{assim}$$

```

globals [
  ;; Filter feeding parameters
  spm-sd                ;; standard deviation for Gaussian SPM preference curve (2
  max-filter-distance   ;; how far fish can evaluate patches (optional)

  ;; Biomagnification constants from Lavoie et al. 2013
  BMF-MeHg              ;; 1.74 (MeHg biomagnification factor)
  BMF-Hg                ;; 1.38 (Hg biomagnification factor)
]

patches-own [
  ;; physical environment
  SPM                  ;; suspended particulate matter
  patch-terrain        ;; "water" or "land"

  ;; contaminant concentrations
  mercury              ;; inorganic Hg
  methylmercury        ;; MeHg
]

fish-own [
  ;; -----
  ;; Filter feeding physiology
  ;; -----
  size                 ;; body size scaling filtration
  metabolism-rate      ;; modulates filtration intensity
  digestion-efficiency  ;; fraction of filtered biomass digested
  optimal-SPM          ;; preferred SPM concentration
  filter-feed?         ;; Boolean: whether agent filter feeds

  ;; -----
  ;; Movement & energy
  ;; -----
  speed                ;; movement rate
  swim-efficiency      ;; metabolism-adjusted efficiency
  E-swim               ;; energy cost per unit distance
  energy               ;; energetic budget

  ;; -----
  ;; Stomach contents
  ;; -----
  stomach-contents
  stomach-capacity
  stomach-contents-Hg  ;; inorganic Hg from filter feeding

```

```

stomach-contents-MeHg          ;; MeHg from filter feeding

;; -----
;; Contamination physiology
;; (needed for risk integration)
;; -----
hg-total                      ;; cumulative Hg body burden
mehg-total                    ;; cumulative MeHg body burden
hg-exposure-total
mehg-exposure-total
hg-exposure-total-normalized
mehg-exposure-total-normalized
]

to filter-feed
if filter-feed? [
  let spm-sd 25
  ;; Base filtration capacity scales with body size *and* metabolism
  let base-filtration ((weight / 100) * metabolism-rate)
  set base-filtration (base-filtration * digestion-efficiency)
  let neighbors-list sort (neighbors with [patch-terrain = "water"])
  let best-patch nobody
  let best-score -999

  foreach neighbors-list [p ->
    let spm-patch [SPM] of p
    let interest exp (- ((spm-patch - optimal-SPM) ^ 2) / (2 * (spm-sd ^ 2)))
    let benefit (spm-patch * interest * base-filtration)
    let comp count alewives-on p
    if comp > 0 [ set benefit benefit / comp ]
    if benefit > best-score [
      set best-score benefit
      set best-patch p
    ]
  ]
]

if best-patch != nobody and [patch-terrain] of best-patch = "water" [
  face best-patch
  face best-patch
  move-to best-patch

  ;; calculate distance between best patch and previous patch
  let travel-distance distancexy previous-x previous-y

  ;; --- Swimming energy cost (scaled by efficiency) ---

```

```

set energy max list 0 (energy - (E-swim * travel-distance))

;; --- Feeding and digestion (temperature-dependent) ---
let patch-spm [SPM] of best-patch
let prey-assimilated patch-spm * base-filtration * 300

;; --- Contaminant uptake (SPM * Hg and MeHg) ---
let spm-hg [mercury] of best-patch
let spm-mehg [methylmercury] of best-patch

;; --- Biomagnification ---
let biomag-factor (random-float 30) + 60 ;; random 60-90
let hg-assimilated (patch-spm * spm-mehg * biomag-factor * prey-assimilated)
let mehg-assimilated (patch-spm * spm-hg * (biomag-factor / 10) * prey-assimilated)

;; Remaining capacity
let space-available (stomach-capacity - stomach-contents)

;; You can only eat as much as fits
let intake (min (list prey-assimilated space-available))

;; Add biomass to stomach
set stomach-contents (stomach-contents + intake)
set stomach-contents-Hg (stomach-contents-Hg + hg-assimilated);; mercury stomach and
set stomach-contents-MeHg (stomach-contents-MeHg + mehg-assimilated);; methylmercury stomach and

]
end

```

9.8 Overview

Lipid catabolism provides a secondary energy pathway that fish use when immediate food resources are unavailable or when energy falls below a functional threshold. In this submodel, agents metabolize stored lipid reserves to restore energy, maintain migration, and avoid starvation. Lipid use is influenced by body condition, metabolic efficiency, and the magnitude of energetic deficit. Lipid depletion also reduces body weight over time, creating a biologically meaningful cost to sustained lipid reliance. Contaminant uptake does not occur during lipid catabolism, reflecting the absence of feeding.

9.9 Purpose

The purpose of this submodel is to simulate endogenous energy mobilization through lipid reserves when fish cannot meet energetic needs through feeding. Lipid catabolism allows migratory fish to continue movement and survive temporary periods of low food availability. This mechanism reflects empirical observations of lipid use during spawning migrations, overwintering, and other energetically demanding life history periods.

This submodel is best used in tandpm with an energy recovery strategy like filter feeding and digestion, or `pref=dation` and `digesiton`

9.10 Entities, State Variables, and Scales

9.10.1 Spatial and Temporal Scales

Spatial Unit: Patch (3 m x 3 m resolution)

Temporal Unit: 5 minute time steps (tick)

9.10.2 Global Variables

Global Variable	Definition
full-energy-level	Target energy level used to compute energetic deficit during lipid use (default = 100).
energy-per-lipid	Energy returned per unit lipid mass mobilized (default = 40), used to convert lipid into energy.
lipid-refill-fraction	Fraction of body mass available to mobilize as lipid during fasting (default = 0.5).
lipid-max-fraction (<i>optional</i>)	Upper bound on lipid fraction that can be lost before mortality or physiological collapse (not currently enforced but biologically relevant).
debug-lipid? (<i>optional</i>)	Toggles printing of intermediate lipid catabolism calculations for debugging.

9.10.3 Patch Variables

Lipid catabolism does not directly depend on patch level variables, since no external resources are consumed.

9.10.4 Agent Variables

Variable Name	Definition
energy	Current energy level of the agent
weight	Current body mass of the fish
original-weight	Initial body mass used to scale maximum lipid availability
lipid-catabolism-efficiency	Efficiency with which lipids are converted into usable energy. This value is equal to the current metabolic rate divided by the metabolic rate at optimal temperature, allowing lipid use to scale with physiological performance
lipid-used	Amount of lipid converted to energy during the current step
metabolism-rate	Current metabolic rate, which influences efficiency scaling
baseline-metabolism-rate	Metabolic rate at optimal temperature
mehg-foraging	Methylmercury acquired during feeding (set to zero when lipids are used)
hg-foraging	Inorganic mercury acquired during feeding (set to zero when lipids are used)
mehg-foraging-total	Total methylmercury obtained from foraging
hg-foraging-total	Total inorganic mercury obtained from foraging
mehg-total	Total methylmercury accumulated
hg-total	Total inorganic mercury accumulated

9.11 Process Overview and Scheduling

1. Compute energy deficit relative to a target energy level of one hundred.
2. Determine the lipid amount needed to fill the deficit.
3. Limit lipid use by the maximum lipid available for mobilization.
4. Apply lipid catabolism efficiency to determine usable lipid energy.

5. Convert mobilized lipid to energy.
6. Reduce weight by the amount of lipid burned.
7. Set contaminant uptake to zero during this process, since no feeding occurs.

9.12 Design Concepts

Basic Principles

Lipid catabolism models endogenous energy mobilization in the absence of food. Fish draw upon lipid stores when energy is low, which is a well documented physiological process during migration, overwintering, and fasting.

Emergence

Patterns of weight loss, energy cycling, and recovery emerge from repeated lipid use during periods of limited food access.

Adaptation

Agents adaptively mobilize lipids based on the severity of their energy deficit and on efficiency scaling from the metabolism submodel.

Objectives

Agents seek to maintain enough energy to migrate, forage, avoid predation, and prevent starvation.

Sensing

Agents sense their own internal energy level and lipid reserve availability.

Interaction

Lipid catabolism does not require environmental interaction and therefore represents a purely endogenous process.

Stochasticity

This submodel is deterministic. However, variation in weight, metabolism, and previous feeding creates emergent differences among individuals.

Observation

Outputs include energy restored through lipid catabolism, lipid used, and body mass loss over time.

9.13 Initialization

Variable	Initialized Value	Justification
weight	species specific	Establishes initial body condition
original-weight	equal to initial weight	Defines total lipid availability
energy	100	Full energy at initialization
lipid-catabolism-efficiency	from metabolism submodel	Reflects physiological scaling of lipid use
lipid-used	0	No lipid used at start

9.14 Submodels

9.14.1 Lipid Mobilization

Maximum lipid available for mobilization depends on relative body mass:

$$Lipid_{refill} = 0.5 \times \left(\frac{weight}{original_weight} \right)$$

Energy deficit relative to full energy:

$$Energy_{deficit} = 100 - Energy$$

Lipid required to refill the deficit:

$$Lipid_{needed} = \frac{Energy_{deficit}}{40}$$

Actual lipid mobilized:

$$Lipid_{used} = \min(Lipid_{needed}, Lipid_{refill}) \times lipid_catabolism_efficiency$$

9.14.2 Energy Conversion

Mobilized lipid is converted into energy:

$$Energy_{added} = Lipid_{used} \times 40$$

Energy is capped at one hundred:

$$Energy = \min(100, Energy + Energy_{added})$$

9.14.3 Weight Loss

Weight decreases proportionally to lipid used:

$$Weight = Weight - Lipid_{used}$$

9.14.4 Contaminant Bookkeeping

No mercury or methylmercury is acquired during lipid catabolism:

$$MeHg_{digested} = 0$$

$$Hg_{digested} = 0$$

These values are added to cumulative totals to maintain bookkeeping consistency, but do not increase contaminant burden.

9.15 Netlogo Implementation

```
;; =====
;;  LIPID CATABOLISM SUBMODEL DECLARATIONS
;;  =====

turtles-own [

  ;; Energetics & body condition
  energy
  weight
  original-weight

  ;; Lipid metabolism
  lipid-catabolism-efficiency
  lipid-used

  ;; Metabolic rate inputs
  metabolism-rate
  baseline-metabolism-rate

  ;; Contaminant bookkeeping during fasting
  mehg-foraging
```

```

hg-foraging

mehg-foraging-total
hg-foraging-total

mehg-total
hg-total
]

;; =====
;;  LIPID CATABOLISM SUBMODEL
;;  =====

to lipid-loss
  ;; Maximum lipid metabolized this step, adjusted by efficiency
  let lipid-refill 0.5 * (weight / original-weight)
  ;print (word "lipid-refill: " lipid-refill)
  ;print (word "current energy: " energy)
  ;; Calculate deficit and lipid need
  let energy-deficit (100 - energy)
  ;print (word "energy-deficit: " energy-deficit)
  let lipid-needed (energy-deficit / 40)
  ;print (word "lipid-needed: " lipid-needed)

  ;; Actual lipid use depends on efficiency and availability
  let lipid-used (min list lipid-needed lipid-refill)
  set lipid-used (lipid-used * lipid-catabolism-efficiency)
  ;print (word "lipid-used: " lipid-used)

  ;; Convert lipid to energy (scaled by efficiency)
  let energy-change (lipid-used * 40)
  ;print (word "energy added: " energy-change)
  set energy (energy + energy-change)
  ;print (word "energy after lipid conversion: " energy)
  if energy > 100 [
    set energy 100
    print "energy capped at 100"
  ]

  ;; Weight loss proportional to lipid burned
  set weight weight - (lipid-used)
  ;print (word "weight after lipid loss: " weight)

  ;; When relying on lipids, Hg mercury foraging risk stops

```

```
;; contaminant bookkeeping
let digested-MeHg 0
let digested-Hg 0
set meh-g-foraging digested-MeHg
set hg-foraging digested-Hg

set meh-g-foraging-total meh-g-foraging-total + digested-MeHg
set hg-foraging-total hg-foraging-total + digested-Hg
set meh-g-total meh-g-total + digested-MeHg
set hg-total hg-total + digested-Hg
end
```


Chapter 10

Landward Migration

10.1 Overview

This function simulates landward migration for fish moving upstream through riverine and estuarine systems. Agents adjust their movement based on local water velocity, energetic condition, and the relative difficulty of swimming in a given flow. Movement occurs along a least cost path that is recalculated according to hydrodynamic conditions.

10.2 Purpose

The purpose of this submodel is to represent upstream migration under realistic spatial hydrodynamic resistance. Agents evaluate flow velocity, movement difficulty, body condition, and patch level travel costs before advancing along a least cost path toward a home location.

10.3 Entities, State Variables, and Scales

10.3.1 Spatial and Temporal Scales

Spatial Unit: Patch (3 m x 3 m resolution)

Temporal Unit: 5 minute time steps (tick)

10.3.2 Global Variables

Variable	Definition
max-seaward-velocity	Maximum seaward directed velocity in the domain, used for normalization.
max-landward-velocity	Maximum landward directed velocity in the domain.

10.3.3 Patch Variables

Variable	Definition
velocity	Depth averaged hydrodynamic velocity at the patch.
depth	Water depth at the patch.
patch-terrain	Indicates whether the patch is water or land.
cost-to-home	Patch level cost field producing a least cost corridor.
visits-by-alewife	Number of times the agent visits the patch.
ticks-spent-alewife	Time spent on the patch.

10.3.4 Agent Variables

Variable	Definition
weight	Body mass of the fish.
speed	Current swimming speed.
prev-speed	Previous time step swimming speed.
max-speed	Maximum sustained speed.
min-speed	Minimum allowable speed.
swim-efficiency	Scaling factor for acceleration and deceleration.
difficulty-factor	Difficulty of movement under local velocity conditions.
planned-path	Sequence of patches in the least cost route.
home-patch	Destination patch for migration.
trail	Record of patches visited by the agent.

10.4 Process Overview and Scheduling

1. Compute swimming difficulty from hydrodynamic velocity and body size.
2. Calculate swimming speed using desired-speed, velocity influence, and acceleration limits.
3. Evaluate the least cost field and construct a path to the home patch.
4. Move along the path for a distance proportional to swimming speed.
5. Record patch visitation and time spent.

10.5 Design Concepts

Basic Principles

Hydrodynamic constraints, difficulty scaling, and body condition influence agent movement.

Emergence

Migration routes, temporal occupancy, and movement speed emerge from repeated interactions of flow and difficulty.

Objectives

Agents attempt to reach the home patch but do not optimize globally. Movement follows the least cost route at each step.

Sensing

Agents sense velocity and depth of the local and neighboring patches.

Observation

Paths, timing, and residency can be monitored chronologically.

10.6 Initialization

Variable	Initial Value	Justification
velocity	user-defined	Hydrodynamic input data.
weight	species-specific	Required for difficulty scaling.
max-speed	species-specific	Physiological constraint.
min-speed	species-specific	Prevents speed collapse.

Variable	Initial Value	Justification
swim-efficiency	species or constant	Controls acceleration rate.
difficulty-factor	1	Neutral starting difficulty.
prev-speed	starting speed	Smooth initial movement.

10.7 Submodels

10.7.1 Swimming Difficulty

Swimming difficulty reflects the hydrodynamic effort required for the agent to move through its current patch. Velocity is normalized between the maximum observed landward and seaward velocities:

$$V_{\text{norm}} = \frac{V_{\text{patch}} - V_{\text{min}}}{V_{\text{max}} - V_{\text{min}}}$$

This normalized velocity is scaled by relative body size $M_{\text{agent}}/M_{\text{max}}$ and raised to a scaling exponent k :

$$Df_{\text{raw}} = \left(\frac{V_{\text{norm}}}{M_{\text{agent}}/M_{\text{max}}} \right)^k$$

Difficulty is then mapped onto the one to ten range:

$$D_f = 1 + 9 \cdot Df_{\text{raw}}$$

and constrained:

$$D_f \in [1, 10]$$

Difficulty increases when:

- M_{agent} is small
- $|V_{\text{patch}}|$ approaches its maximum value
- flow opposes movement

10.7.2 Swimming Speed

Swimming speed depends on energetic condition, hydrodynamic difficulty, and the effect of local flow. The agent first computes an energy factor:

$$E_{\text{factor}} = \frac{E_{\text{agent}}}{100}$$

Velocity impact from hydrodynamics is:

$$V_{\text{impact}} = k \cdot V_{\text{patch}} \cdot 300$$

The desired swimming speed is:

$$V_{\text{desired}} = \frac{V_{\text{max}} \cdot E_{\text{factor}}}{D_f} + V_{\text{impact}}$$

Speed is constrained between a minimum and maximum:

$$V_{\text{desired}} = \min(V_{\text{max}}, \max(V_{\text{min}}, V_{\text{desired}}))$$

Acceleration is smoothed using the previous speed:

$$\Delta V_{\text{max}} = 0.5 \cdot \text{swim_efficiency}$$

If $|V_{\text{desired}} - V_{\text{prev}}| > \Delta V_{\text{max}}$, then speed moves gradually toward the desired value. Otherwise speed is set directly.

10.7.3 Path Based Movement

Landward migration uses a least cost routing strategy instead of direct vector movement. The number of patches traversed in a tick is based on swimming speed:

$$\text{travel_distance} = \frac{V_{\text{current}}}{3}$$

Movement follows:

1. compute cost-to-home
2. generate a path using decreasing cost

3. move through the first $\lceil \text{travel_distance} \rceil$ patches
4. record patch visits and time spent
5. append patch-here to the trail if new

10.7.4 Travel Cost

Travel cost for a candidate patch incorporates velocity magnitude, slope, and difficulty:

$$\text{cost}_{\text{raw}} = |V_{\text{patch}}| + 0.5 \cdot \max(0, \Delta\text{depth})$$

Difficulty multiplies the base cost:

$$\text{cost} = \text{cost}_{\text{raw}} \cdot D_f$$

This produces a cost-to-home field representing hydrodynamic resistance.

10.7.5 Swimming Energy

Swimming incurs a metabolic cost that scales with base swimming cost, difficulty, and swim efficiency.

Let $\beta = 0.75$.

Let the effective multiplier be:

$$M = \frac{D_f}{\text{swim_efficiency}}$$

Where lower efficiency results in higher energetic cost.

Energy expenditure is:

$$E_{\text{swim}} = \text{swim_base} \cdot M^\beta$$

Total energy is reduced accordingly:

$$E_{\text{agent}} = \max(0, E_{\text{agent}} - E_{\text{swim}})$$

Higher difficulty and lower swim efficiency increase metabolic cost.

```

;; =====
;; LANDWARD MIGRATION + HYDRODYNAMIC SWIMMING SUBMODEL DECLARATIONS
;; =====

globals [

  ;; Velocity normalization bounds
  max-seaward-velocity
  max-landward-velocity
]

patches-own [

  ;; Environmental attributes
  velocity
  depth
  patch-terrain    ;; "water" or "land"

  ;; Least-cost routing fields
  cost-to-home
  cost-to-sea
  cost-to-prey

  ;; Tracking fish patch-level behavior
  visits-by-fish
  ticks-spent-fish
]

turtles-own [

  ;; Body condition
  weight

  ;; Swimming behavior
  speed
  prev-speed
  max-speed
  min-speed
  swim-efficiency

  ;; Hydrodynamic resistance
  difficulty-factor

  ;; Energetics
  E-swim

```

```

swim-base

;; Migration targets
home-patch
migration-patch

;; Movement memory
planned-path
trail
previous-patch
previous-x
previous-y
]

;; =====
;;  SWIMMING DIFFICULTY (LANDWARD)
;;  =====

to calculate-difficulty-landward

  let M-max max [weight] of breed      ;; largest fish in population
  let M-agent weight                    ;; focal individual
  let V-max max-seaward-velocity
  let V-min max-landward-velocity
  let k 0.75                            ;; scaling exponent

  ;; Normalize velocity
  let normalized-velocity (velocity - V-min) / (V-max - V-min)

  ;; Difficulty from normalized velocity + size ratio
  let Df-raw (normalized-velocity / (M-agent / M-max)) ^ k

  ;; Scale to range 1-10
  set difficulty-factor (1 + 9 * Df-raw)
  set difficulty-factor max list 1 (min list 10 difficulty-factor)
end

;; =====
;;  SWIMMING SPEED (LANDWARD)
;;  =====

to calculate-swimming-speed-landward

```

```

let k -0.75                                ;; velocity influence
let energy-factor (energy / 100)           ;; 0-1 scale
let velocity-impact (k * velocity * 300)   ;; 5-min timestep scaling

;; Preliminary target speed
let desired-speed
  (max-speed * energy-factor / difficulty-factor)
  + velocity-impact

;; Clamp desired speed to biologically realistic bounds
set desired-speed min list max-speed (max list min-speed desired-speed)

;; Smooth acceleration / deceleration
let max-speed-change 0.5
set max-speed-change (max-speed-change * swim-efficiency)

if abs(desired-speed - prev-speed) > max-speed-change [
  if desired-speed > prev-speed [
    set speed prev-speed + max-speed-change
  ]
  if desired-speed < prev-speed [
    set speed prev-speed - max-speed-change
  ]
]

if abs(desired-speed - prev-speed) <= max-speed-change [
  set speed desired-speed
]

set prev-speed speed
end

;; =====
;; LEAST-COST ROUTING TO HOME (LANDWARD)
;; =====

to calculate-cost-to-home [homing-patch]

  ;; initialize routing fields
  ask home-patch [ set cost-to-home 0 ]
  ask patch-here [ set cost-to-home 1e9 ]

  let frontier (list home-patch)
  let df difficulty-factor                ;; hydrodynamic penalty

```

```

while [not empty? frontier] [
  let current first frontier
  set frontier but-first frontier

  ask current [
    let current-cost cost-to-home

    ask neighbors4 with [patch-terrain = "water"] [

      let travel-cost compute-travel-cost current self df
      let new-cost current-cost + travel-cost

      if new-cost < cost-to-home [
        set cost-to-home new-cost
        set frontier lput self frontier
      ]
    ]
  ]
]
end

;; =====
;; BUILD OPTIMAL PATH ENTIRELY FROM COST-FIELD
;; =====

to-report build-path [current-patch homing-patch]

  let path (list current-patch)
  let cur current-patch
  let max-steps 2000

  while [cur != homing-patch and max-steps > 0] [
    set max-steps max-steps - 1

    let candidates [neighbors4] of cur
    set candidates candidates with [
      patch-terrain = "water" and
      cost-to-home < [cost-to-home] of cur
    ]

    if not any? candidates [
      report path
    ]
  ]

```

```

    let next-patch min-one-of candidates [cost-to-home]
    set path lput next-patch path
    set cur next-patch
  ]

  report path
end

;; =====
;; TRAVEL COST UNDER HYDRODYNAMIC RESISTANCE
;; =====

to-report compute-travel-cost [from-patch to-patch df]

  ;; velocity resistance
  let raw-vel [velocity] of to-patch
  let velocity-resistance
    ifelse-value is-number? raw-vel [abs raw-vel] [0]

  ;; depth slope component
  let from-level [depth] of from-patch
  let to-level [depth] of to-patch
  let slope 0

  if (is-number? from-level and is-number? to-level) [
    set slope max list 0 (to-level - from-level)
  ]

  ;; hydrodynamic cost + slope penalty
  let base-cost velocity-resistance + (0.5 * slope)

  ;; difficulty multiplier
  let df2 ifelse-value is-number? df [df] [1]

  report max list 0 (base-cost * df2)
end

;; =====
;; LANDWARD MIGRATION - PATH EXECUTION
;; =====

to migrate-landward

```

```

let travel-distance (speed / 3)
let path build-path patch-here home-patch
set planned-path path

if not empty? path [

  let steps-to-move ceiling travel-distance
  let move-patches sublist path 0 (min list steps-to-move length path)

  let old-patch patch-here

  foreach move-patches [ p ->

    ;; Record patch visitation
    if p != old-patch [
      ask p [ set visits-by-alewife visits-by-alewife + 1 ]
      set old-patch p
    ]

    ;; Move into patch
    face p
    move-to p

    ask p [
      set ticks-spent-alewife ticks-spent-alewife + 1
    ]
  ]

  ;; Add patch to historical trail
  if not member? patch-here trail [
    set trail lput patch-here trail
  ]
end

;; =====
;; UNIVERSAL SWIMMING ENERGY COST
;; =====

to calculate-swim-energy

  let beta 0.75

  let energy-multiplier

```



```
      (difficulty-factor * (1 / swim-efficiency))

  set E-swim
    (swim-base * (energy-multiplier ^ beta))

  set energy max list 0 (energy - E-swim)
end
```


Chapter 11

Seaward Migration

11.1 Overview

This function simulates seaward migration for fish moving downstream through riverine and estuarine systems. Agents respond to hydrodynamic conditions by adjusting swimming difficulty, swimming speed, and movement along a least cost pathway toward a downstream migration patch.

11.2 Purpose

The purpose of this submodel is to represent downstream migration under realistic hydrodynamic resistance. Agents evaluate flow velocity, movement difficulty, energetic condition, and travel costs before advancing along a least cost route toward a migration destination.

11.3 Entities, State Variables, and Scales

11.3.1 Spatial and Temporal Scales

Spatial Unit: Patch (3 m x 3 m resolution)

Temporal Unit: 5 minute time steps (tick)

11.3.2 Global Variables

Variable	Definition
max-seaward-velocity	Maximum observed seaward velocity in the domain.
max-landward-velocity	Maximum observed landward velocity in the domain.

11.3.3 Patch Variables

Variable	Definition
velocity	Depth averaged hydrodynamic velocity.
depth	Water depth at the patch.
patch-terrain	Indicates whether the patch is land or water.
cost-to-sea	Patch-level travel cost for seaward movement.
visits-by-alewife	Number of times the agent visits the patch.
ticks-spent-alewife	Time spent on the patch.

11.3.4 Agent Variables

Variable	Definition
weight	Body mass of the fish.
speed	Current swimming speed.
prev-speed	Previous time step swimming speed.
max-speed	Maximum sustained speed.
min-speed	Minimum allowable speed.
swim-efficiency	Swimming acceleration and energy scaling.
difficulty-factor	Hydrodynamic difficulty.
planned-path	Sequence of patches in least cost route.
migration-patch	Destination patch.
trail	Record of visited patches.

11.4 Process Overview and Scheduling

1. Calculate swimming difficulty using normalized velocity and body size.
2. Update swimming speed using energy reserves, difficulty, and flow velocity.
3. Calculate cost-to-sea and construct least cost path.

4. Move along the first part of the path based on swimming speed.
5. Record patch visits and time spent.

11.5 Design Concepts

Basic Principles

Hydrodynamic forcing, body size, and energy availability determine downstream movement.

Emergence

Route choice, travel duration, and energetic depletion emerge from repeated state updates.

Objectives

Agents attempt to reach the seaward migration patch following a hydrodynamically efficient route.

Sensing

Agents sense local velocity, slope, difficulty, and patch cost gradients.

Observation

Movement trajectories, energy decline, and convergence routes can be monitored.

11.6 Initialization

Variable	Initial Value	Justification
velocity	Hydrodynamic input	Required environmental forcing.
weight	Species-specific	Required for difficulty scaling.
max-speed	Species-specific	Physiological constraint.
min-speed	Species-specific	Prevents collapse to zero speed.
swim-efficiency	Species or constant	Controls acceleration and deceleration.
difficulty-factor	1	Neutral start value.
prev-speed	Starting speed	Smooth initial movement.

11.7 Submodels

11.7.1 Swimming Difficulty

Velocity is normalized between minimum and maximum observed values:

$$V_{\text{norm}} = \frac{V_{\text{patch}} - V_{\text{min}}}{V_{\text{max}} - V_{\text{min}}}$$

Difficulty scales with body size:

$$Df_{\text{raw}} = \left(\frac{V_{\text{norm}}}{M_{\text{agent}}/M_{\text{max}}} \right)^k$$

Difficulty is mapped onto the one-to-ten range:

$$D_f = 1 + 9 \cdot Df_{\text{raw}}$$

and constrained:

$$D_f \in [1, 10]$$

11.7.2 Swimming Speed

Energy factor:

$$E_{\text{factor}} = \frac{E_{\text{agent}}}{100}$$

Flow velocity effect:

$$V_{\text{impact}} = k \cdot V_{\text{patch}} \cdot 300$$

Desired swimming speed:

$$V_{\text{desired}} = \frac{V_{\text{max}} \cdot E_{\text{factor}}}{D_f} + V_{\text{impact}}$$

Speed constraints:

$$V_{\text{desired}} = \min(V_{\text{max}}, \max(V_{\text{min}}, V_{\text{desired}}))$$

Acceleration smoothing:

$$\Delta V_{\max} = 0.5 \cdot \text{swim_efficiency}$$

11.7.3 Path-Based Movement

Travel distance:

$$\text{travel_distance} = \frac{V_{\text{current}}}{3}$$

Movement follows:

1. compute cost-to-sea
2. build path using decreasing cost
3. traverse first $\lceil \text{travel_distance} \rceil$ patches
4. record visitation and time spent
5. append to the movement trail

11.7.4 Travel Cost

Raw cost:

$$\text{cost}_{\text{raw}} = |V_{\text{patch}}| + 0.5 \cdot \max(0, \Delta \text{depth})$$

Difficulty scaling:

$$\text{cost} = \text{cost}_{\text{raw}} \cdot D_f$$

11.7.5 Swimming Energy

Let $\beta = 0.75$.

Energy multiplier:

$$M = \frac{D_f}{\text{swim_efficiency}}$$

Energy cost:

$$E_{\text{swim}} = \text{swim_base} \cdot M^{\beta}$$

Energy update:

$$E_{\text{agent}} = \max(0, E_{\text{agent}} - E_{\text{swim}})$$

11.8 NetLogo Implementation

```
;; =====
;; LANDWARD MIGRATION + HYDRODYNAMIC SWIMMING SUBMODEL DECLARATIONS
;; =====

globals [

  ;; Velocity normalization bounds
  max-seaward-velocity
  max-landward-velocity
]

patches-own [

  ;; Environmental attributes
  velocity
  depth
  patch-terrain    ;; "water" or "land"

  ;; Least-cost routing fields
  cost-to-home
  cost-to-sea
  cost-to-prey

  ;; Tracking fish patch-level behavior
  visits-by-fish
  ticks-spent-fish
]

turtles-own [

  ;; Body condition
  weight
```



```

;; Swimming behavior
speed
prev-speed
max-speed
min-speed
swim-efficiency

;; Hydrodynamic resistance
difficulty-factor

;; Energetics
E-swim
swim-base

;; Migration targets
home-patch
migration-patch

;; Movement memory
planned-path
trail
previous-patch
previous-x
previous-y
]

;; =====
;; SEAWARD MIGRATION + HYDRODYNAMIC SWIMMING SUBMODEL
;; =====

;; -----
;; Swimming Difficulty (Seaward)
;; -----
to calculate-difficulty-seaward

  let M-max max [weight] of breed
  let M-agent weight
  let V-max max-landward-velocity
  let V-min max-seaward-velocity
  let k 0.75

  let normalized-velocity (velocity - V-min) / (V-max - V-min)

  let Df-raw (normalized-velocity / (M-agent / M-max)) ^ k

```

```

    set difficulty-factor (1 + 9 * Df-raw)
    set difficulty-factor max list 1 (min list 10 difficulty-factor)
end

```

```

;; -----
;;  Swimming Speed (Seaward)
;; -----

```

```

to calculate-swimming-speed-seaward

```

```

    let k 0.75
    let energy-factor (energy / 100)

    let velocity-impact (k * velocity * 300)

```

```

    let desired-speed
      (max-speed * energy-factor / difficulty-factor)
      + velocity-impact

```

```

    set desired-speed
      min list max-speed (max list min-speed desired-speed)

```

```

    let max-speed-change 0.5
    set max-speed-change (max-speed-change * swim-efficiency)

```

```

    if abs(desired-speed - prev-speed) > max-speed-change [
      if desired-speed > prev-speed [
        set speed prev-speed + max-speed-change
      ]
      if desired-speed < prev-speed [
        set speed prev-speed - max-speed-change
      ]
    ]

```

```

    if abs(desired-speed - prev-speed) <= max-speed-change [
      set speed desired-speed
    ]

```

```

    set prev-speed speed
end

```

```

;; -----
;;  Cost-to-Sea (Least Cost Routing)
;; -----

```

```

to calculate-cost-to-sea [sea-patch]

  ask migration-patch [ set cost-to-sea 0 ]
  ask patch-here      [ set cost-to-sea 1e9 ]

  let frontier (list migration-patch)
  let df difficulty-factor

  while [not empty? frontier] [

    let current first frontier
    set frontier but-first frontier

    ask current [

      let current-cost cost-to-sea

      ask neighbors4 with [patch-terrain = "water"] [

        let travel-cost compute-travel-cost current self df
        let new-cost current-cost + travel-cost

        if new-cost < cost-to-sea [
          set cost-to-sea new-cost
          set frontier lput self frontier
        ]
      ]
    ]
  ]
end

;; -----
;; Build Path to Sea
;; -----
to-report build-path-sea [current-patch sea-patch]

  let path (list current-patch)
  let cur current-patch
  let max-steps 2000

  while [cur != migration-patch and max-steps > 0] [

    set max-steps max-steps - 1

```

```

    let candidates [neighbors] of cur
    set candidates candidates with [
      patch-terrain = "water" and
      cost-to-sea < [cost-to-sea] of cur
    ]

    if not any? candidates [
      report path
    ]

    let next-patch min-one-of candidates [cost-to-sea]
    set path lput next-patch path
    set cur next-patch
  ]

  report path
end

;; -----
;; Seaward Migration - Movement Execution
;; -----
to migrate-seaward

  ;; Schooling logic (excluded in documentation)
  find-schoolmates
  if any? schoolmates [
    find-nearest-neighbor
    ifelse distance nearest-neighbor < minimum-separation [
      separate
    ] [
      cohere
      align
    ]
    adjust-speed
  ]

  ;; Path-based movement
  let travel-distance (speed / 3)
  let path build-path-sea patch-here migration-patch
  set planned-path path

  if not empty? path [

    let steps-to-move ceiling travel-distance

```

```

    let move-patches sublist path 0 (min list steps-to-move length path)

    let old-patch patch-here

    foreach move-patches [ p ->

      if p != old-patch [
        ask p [ set visits-by-alewife visits-by-alewife + 1 ]
        set old-patch p
      ]

      face p
      move-to p

      ask p [
        set ticks-spent-alewife ticks-spent-alewife + 1
      ]
    ]

    if not member? patch-here trail [
      set trail lput patch-here trail
    ]
  end

;; -----
;; Universal Swimming Energy Cost
;; -----
to calculate-swim-energy

  let beta 0.75
  let energy-multiplier (difficulty-factor * (1 / swim-efficiency))

  set E-swim (swim-base * (energy-multiplier ^ beta))
  set energy max list 0 (energy - E-swim)
end

```


Chapter 12

Schooling

12.1 Overview

Schooling is an emergent collective behavior in which fish align, cohere, and maintain spacing with nearby con-specifics. In this model, schooling increases hydrodynamic efficiency, predator avoidance, and coordinated migration. The submodel follows the classic behavioral rules from agent based schooling and BOIDS frameworks (separation, alignment, and cohesion).

The schooling behavior in this model is adapted from the publicly available NetLogo model Fish Schooling For Predator Avoidance created by Sebastian Dobon and shared through the NetLogo Modeling Commons. That model implements classic schooling rules from the original BOIDS framework developed by Craig Reynolds in 1987 in his paper *Flocks, Herds, and Schools: A Distributed Behavioral Model* [Reynolds].

12.2 Purpose

The purpose of this submodel is to simulate collective movement patterns observed in migratory fish where individuals adjust their position and speed based on nearby schoolmates. This mechanism enhances group cohesion during landward or seaward migration and produces emergent school structure, directional alignment, and spacing.

12.3 Entities, State Variables, and Scales

12.3.1 Spatial and Temporal Scales

Spatial Unit: Patch (3 m x 3 m resolution)

Temporal Unit: 5 minute time steps (tick)

12.3.2 Patch Variables

Variable Name	Definition
patch-terrain	Identifies water or land patches

12.3.3 Agent Variables

Variable Name	Definition
breed	Species identity used to restrict schooling to con-specifics
speed	Current swimming speed
min-speed	Lowest allowable swimming speed
start-migration?	Indicates whether the agent has begun migration
landward-migration?	Indicates landward migration mode
seaward-migration?	Indicates seaward migration mode
schoolmates	Set of nearby con-specifics that meet schooling criteria
nearest-neighbor	Closest schoolmate used for spacing and cohesion
minimum-separation	Desired minimum inter-fish spacing

12.4 Process Overview and Scheduling

1. Identify schoolmates based on species identity and shared migration state.
2. Detect the nearest neighbor among schoolmates.
3. Apply separation when neighbors are too close.
4. Apply cohesion and alignment when spacing is adequate.
5. Adjust speed to match group speed and maintain coordination.

12.5 Design Concepts

Basic Principles

Schooling arises from separation, cohesion, and alignment. These rules reflect collective behaviors observed in migrating fishes.

Emergence

Cohesive schools, spacing patterns, and synchronized direction arise without centralized control.

Adaptation

Agents adjust heading and speed in response to nearby schoolmates.

Objectives

Fish do not attempt to form schools explicitly. Schooling emerges from local rules.

Learning

There is no internal learning. Behavior is reactive and rule based.

Prediction

Agents do not predict future states of neighbors. They respond to current spacing and direction.

Sensing

Agents sense species identity, distance to schoolmates, spacing, and local movement direction.

Interaction

Agents change speed and heading based on interactions with neighbors.

Stochasticity

Small random turning angles elsewhere in the model introduce variability in movement.

Collectives

Schools form as dynamic collectives with shared direction and spacing rules.

Observation

School size, spacing, direction, and cohesion may be tracked to evaluate emergent schooling patterns.

12.6 Initialization

Variable	Initialized Value	Justification
speed	species specific	Starting swimming speed
min-speed	species specific	Prevents schools from stopping entirely

Variable	Initialized Value	Justification
schoolmates	empty set	No neighbors until movement begins
nearest-neighbor	none	Determined dynamically during the simulation
minimum-separation	species specific	Prevents collisions and maintains spacing

12.7 Submodels

12.7.1 Identifying Schoolmates

Fish select schoolmates from nearby con-specifics that share the same migration state, including the start of migration and direction (landward or seaward).

12.7.2 Nearest Neighbor

The closest schoolmate is used to determine whether spacing rules or cohesion rules should be applied.

12.7.3 Separation

If the nearest neighbor is closer than the minimum desired separation, the agent turns away to maintain spacing.

12.7.4 Cohesion

When spacing is appropriate, the agent turns toward the group centroid to maintain cohesive structure.

12.7.5 Alignment

The agent aligns its heading with the mean direction of nearby con-specifics to maintain synchronized movement.

12.7.6 Speed Adjustment

Speed is regulated to preserve cohesion within the school:

- The agent accelerates if any schoolmate is significantly faster.
- Otherwise, it gradually slows toward the minimum speed.

This produces emergent schooling speeds and prevents fragmentation.

12.8 Netlogo Implementation

```
;; -----
;; SCHOOLING SUBMODEL - VARIABLE DECLARATIONS
;; -----

patches-own [
  ;; Terrain classification
  patch-terrain      ;; "water" or "land"
]

turtles-own [
  ;; -----
  ;; Schooling state variables
  ;; -----
  schoolmates        ;; agentset of nearby con-specifics
  nearest-neighbor    ;; closest fish in schoolmates
  minimum-separation  ;; desired distance to avoid collisions

  ;; -----
  ;; Migration gating used for schooling eligibility
  ;; (schoolmates only form among agents sharing these states)
  ;; -----
  start-migration?    ;; has the agent initiated migration?
  landward-migration? ;; migrating upstream?
  seaward-migration?  ;; migrating downstream?

  ;; -----
  ;; Swimming parameters affected by schooling
  ;; -----
  speed               ;; current swimming speed
  min-speed           ;; lower speed limit to prevent stall
]

;; -----
;; SCHOOLING: MAIN CONTROLLER
;; -----
```

```

to school
  find-schoolmates

  if any? schoolmates [
    find-nearest-neighbor

    ;; Separation rule
    ifelse distance nearest-neighbor < minimum-separation
    [
      separate
    ]
    [
      ;; Cohesion + alignment rules
      cohere
      align
    ]

    ;; Adjust speed to maintain cohesion
    adjust-speed
  ]
end

;; -----
;; FIND SCHOOLMATES (CON-SPECIFICS + SAME MIGRATION MODE)
;; -----

to find-schoolmates
  set schoolmates other turtles with [
    breed = [breed] of myself and
    start-migration? and
    landward-migration? = [landward-migration?] of myself and
    seaward-migration? = [seaward-migration?] of myself
  ]
end

;; -----
;; FIND NEAREST NEIGHBOR
;; -----

to find-nearest-neighbor
  set nearest-neighbor min-one-of schoolmates [ distance myself ]
end

```

```

;; -----
;;  SEPARATION RULE (AVOID COLLISION)
;; -----

to separate
  ;; Turn away from nearest neighbor
  let dx (dx-of nearest-neighbor)
  let dy (dy-of nearest-neighbor)
  rt (random 20 + 160)
  fd 0.1
end

;; -----
;;  COHESION RULE (TURN TOWARD GROUP)
;; -----

to cohere
  if any? schoolmates [
    face mean-heading-of schoolmates
  ]
end

;; -----
;;  ALIGNMENT RULE (MATCH GROUP HEADING)
;; -----

to align
  if any? schoolmates [
    set heading (heading + (mean [heading] of schoolmates - heading) * 0.1)
  ]
end

;; -----
;;  SPEED ADJUSTMENT (MAINTAIN COHESION)
;; -----

to adjust-speed
  ifelse max [speed] of schoolmates > speed + 0.1
  [
    ;; Accelerate to keep up with group
    set speed speed + 0.1
  ]

```

```
[  
  ;; Slow down toward minimum speed  
  if speed > min-speed [  
    set speed speed - 0.2  
  ]  
]  
end
```

Chapter 13

Selective Tidal Stream Transport

13.1 Overview

Selective Tidal Stream Transport (STST) is a behavioral strategy that enables agents to conserve energy by passively drifting with the current. It is triggered when the along-channel velocity of the patch exceeds the agent's effective swimming speed, and that speed is below a species-specific minimum threshold. Once engaged, agents align with the tidal current and are carried downstream or upstream, depending on flow velocity. STST reduces the metabolic cost of movement by substituting active swimming with passive transport. This behavior persists for a limited duration or until swimming ability improves, after which agents resume directional migration.

13.2 Purpose

To simulate a passive energy-conserving behavior in migratory fish that allows them to use tidal currents to move when swimming capacity is insufficient to overcome flow velocities.

13.3 Entities, State Variables, and Scales

13.3.1 Spatial and Temporal Scales

Spatial Unit: Patch (3 m x 3 m resolution)

Temporal Unit: 5 minute time steps (tick)

13.3.2 Global Variables

Variable Name	Definition
max-seaward-velocity	Highest magnitude seaward-directed hydrodynamic velocity across the spatial domain.
max-landward-velocity	Highest magnitude landward-directed hydrodynamic velocity across the spatial domain.

13.3.3 Patch Variables

Variable Name	Definition
Velocity V_{patch}	The along-channel velocity of a given patch, derived from hydrodynamic model inputs, where positive values are in the landward direction and negative values are in the seaward direction.
tidal-transport-in-patch	Count of agents exhibiting tidal stream transport within a patch (for habitat quality analysis).

13.3.4 Agent Variables

Variable Name	Definition
energy E_{agent}	Total energy available to the agent.
swimming energy E_{swim}	Energy expenditure from movement per time step.
base swim energy $swim_{base}$	Baseline energy cost of movement.
swimming difficulty D_f	Velocity-based proxy representing hydrodynamic resistance.

in-STST? $STST_?$	Boolean value indicating if the agent is actively in STST.
swimming speed V_{agent}	The effective swimming speed of the agent.
minimum threshold speed $Speed_{min}$	The minimum speed at which an agent will move.

13.4 Process Overview and Scheduling

1. Compare swimming speed (V_{agent}) with flow speed (V_{patch}).
2. If $|V_{patch}| > V_{agent}$ and $V_{agent} \leq Speed_{min}$, enter STST.
3. In STST: align with current, update position via drift, apply reduced energy cost.
4. If $V_{agent} > Speed_{min}$, exit STST and resume active swimming.

13.5 Design Concepts

Basic Principles: Selective tidal stream transport is based on behavioral ecology and energetics, simulating the tradeoff between active swimming and energy conservation through passive transport.

Emergence: Passive drift behavior and resulting migration paths emerge from agent-flow interactions and individual swimming limitations.

Adaptation: Agents adapt their mode of movement based on their swimming ability relative to environmental flow, dynamically choosing energy-efficient strategies.

Objectives: Agents seek to minimizing energy loss in strong flows.

Sensing: Agents sense their own V_{agent} and the V_{patch} to determine whether passive drift is needed.

Observation: Records STST patch events, energy expenditure (E_{agent}), and displacement are logged to analyze behavior across flow regimes.

13.6 Initialization

Variable	Initialized Value	Justification
V_{agent}	Based on size, energy, and difficulty factor	Reflects agent's swimming capability based on metabolic limits.
$swim_{min}$	Species-specific parameter	Represents the minimum sustained swimming velocity of the agent.
$swim_{max}$	Species-specific parameter	Represents the maximum sustained swimming velocity of the agent.
E_{agent}	100	Assumes full energy at the start of simulation or at spawning.
$swim_{base}$	$0.02 \cdot \frac{M_{agent}^k}{M_{max}}$	Scales locomotion cost nonlinearly with size; can be calibrated.

13.7 Submodels

13.7.1 Trigger Conditions for STST

Agents compare their swimming ability to the flow conditions. If local flow exceeds their capability and their effort is below a defined threshold, they enter STST:

$$|V_{patch}| > V_{agent} \quad \text{and} \quad V_{agent} \leq Speed_{min}$$

While in STST:

Heading aligns with the current (drift vector) & swimming speed is set to:

$$V_{agent} = |V_{patch}|$$

Energy is set as:

$$E_{swim} = swim_{base}$$

Where:

- V_{agent} is the current swimming speed of the agent.
- V_{patch} is the along-channel velocity at the agent's current patch.
- $Speed_{min}$ is the minimum sustainable swimming speed of the agent.
- $swim_{base}$ is the base swimming cost based on agent size.
- E_{swim} is the total energy cost during passive movement.

13.7.2 Behavior During STST

While in STST, agents align with the current (either landward or seaward) and are passively transported:

$$\vec{Y}_{t+1} = \vec{Y}_t + |V_{patch}| \cdot \hat{u}$$

Swimming speed is overwritten:

$$V_{agent} = |V_{patch}|$$

Energy cost is minimized:

$$E_{swim} = swim_{base}$$

Where:

- \vec{Y}_t is the agent's current spatial position.
- \vec{Y}_{t+1} is the position after drifting.
- V_{patch} is the along-channel velocity at the agent's current patch.
- \hat{u}_{patch} is the direction of the patch velocity (unit vector).
- $swim_{base}$ is the base swimming cost based on agent size.
- E_{swim} is the total energy cost during passive movement.

13.7.3 Stop Conditions for STST

Agents exit STST when they regain sufficient swimming capacity to exceed threshold:

$$V_{agent} > Speed_{min}$$

Where:

- V_{agent} is the current swimming speed of the agent.
- $Speed_{min}$ is the minimum sustainable swimming speed of the agent.

This triggers a return to active migratory movement and deactivates $STST_7$.

13.8 Netlogo Implementation

```
;; -----
;;  GLOBALS
;;  -----
globals [
  max-seaward-velocity
  max-landward-velocity

  landward-migration?
  seaward-migration?

  selective-tidal-transport?
  STST-start-tick
]

;; -----
;;  PATCH VARIABLES
;;  -----
patches-own [
  velocity
  depth
  patch-terrain

  tidal-transport-in-patch
]
```

```

;; -----
;;  TURTLE VARIABLES
;; -----
turtles-own [
    speed
    min-speed
    swim-efficiency
    swim-base
    difficulty-factor
    energy
    E-swim

    selective-tidal-transport?
    STST-start-tick
]

;; -----
;;  STST - LANDWARD
;; -----
to selective-tidal-stream-transport-landward
    let V-patch [velocity] of patch-here
    let S-swim (speed / 300) ;; 5 minute speed

    ;; Continue drifting (1 tick cooldown)
    if selective-tidal-transport? and (ticks - STST-start-tick < 1) [
        let drift-distance abs V-patch * (1 - swim-efficiency)
        let drift-target one-of neighbors with [patch-terrain = "water"]
        if drift-target != nobody [ safe-move drift-target drift-distance ]
        set tidal-transport-in-patch tidal-transport-in-patch + 1
        stop
    ]

    ;; Enter STST
    if abs V-patch > S-swim and S-swim <= min-speed [
        set selective-tidal-transport? true
        set STST-start-tick ticks
        let drift-distance abs V-patch * (1 - swim-efficiency)
        let drift-target one-of neighbors with [patch-terrain = "water"]
        if drift-target != nobody [ safe-move drift-target drift-distance ]
        set tidal-transport-in-patch tidal-transport-in-patch + 1
        stop
    ]

    ;; Exit STST → resume active migration
    if S-swim > min-speed or (ticks - STST-start-tick >= 1) [

```

```

    set selective-tidal-transport? false
    calculate-swim-energy
    set energy max list 0 (energy - E-swim)
    migrate-landward
  ]
end

;; -----
;; STST - SEAWARD
;; -----

to selective-tidal-stream-transport-seaward
  let V-patch [velocity] of patch-here
  let S-swim (speed / 300)

  ;; Continue STST drift
  if selective-tidal-transport? and (ticks - STST-start-tick < 1) [
    let drift-distance abs V-patch * (1 - swim-efficiency)
    let drift-target one-of neighbors with [patch-terrain = "water"]
    if drift-target != nobody [ safe-move drift-target drift-distance ]
    set tidal-transport-in-patch tidal-transport-in-patch + 1
    stop
  ]

  ;; Enter STST
  if abs V-patch > S-swim and S-swim <= min-speed [
    set selective-tidal-transport? true
    set STST-start-tick ticks
    let drift-distance abs V-patch * (1 - swim-efficiency)
    let drift-target one-of neighbors with [patch-terrain = "water"]
    if drift-target != nobody [ safe-move drift-target drift-distance ]
    set tidal-transport-in-patch tidal-transport-in-patch + 1
    stop
  ]

  ;; Exit STST → resume seaward migration
  if S-swim > min-speed or (ticks - STST-start-tick >= 1) [
    set selective-tidal-transport? false
    calculate-swim-energy
    set energy max list 0 (energy - E-swim)
    migrate-seaward
  ]
end

;; -----
;; SAFE MOVEMENT HANDLER

```

```
;; -----  
to safe-move [target-patch move-speed]  
  face target-patch  
  let next-patch patch-ahead move-speed  
  
  ifelse [patch-terrain] of next-patch = "water" [  
    fd move-speed  
  ] [  
    let nearby-water one-of patches in-radius 1 with [patch-terrain = "water"]  
    if nearby-water != nobody [ move-to nearby-water ]  
  ]  
end
```


Chapter 14

Predation and Fleeing Behavior

14.1 Overview

This module simulates predator–prey interactions between migratory fish species, specifically predators (i.e., striped bass) and prey (i.e., alewife), during key migratory and staging periods. Predators detect, pursue, and consume prey based on visibility conditions and prey size constraints. Prey detect approaching predators through visual and social cues and respond with directional fleeing behaviors, which vary based on their size, age, energy level, and water clarity. These behaviors reflect biologically observed trade-offs between escape speed, reaction time, and group-based vigilance. The model also tracks predator energy gain and prey energy depletion and recovery, allowing for analysis of individual fitness and risk exposure under different environmental and behavioral conditions.

14.2 Purpose and Patterns

This submodel captures ecologically grounded predator–prey dynamics by simulating behaviorally realistic rules of pursuit and evasion observed in estuarine fishes.

- Prey initiate fleeing based on predator proximity, visual detection, and turbidity-adjusted reaction time, reflecting sensory limitations in murky estuarine water.

- Predators only pursue prey that are within their gape limit and provide a favorable energy return, consistent with optimal foraging theory and size-selective predation.
- Fleeing speed is scaled by energy and body size, capturing the biologically observed trade-off between fatigue and escape performance.
- Directional fleeing (left, right, up, or down) is based on the predator's relative position, aligning with empirical studies on spatial escape responses.
- Patch-level alarm cues propagate fleeing behavior across prey groups, mimicking the benefits of schooling and collective vigilance in fish.
- Energetic constraints influence both predator foraging frequency and prey escape success, contributing to variability in encounter outcomes.
- Spatial tracking of predation events allows identification of high-risk zones, offering insight into how environmental conditions shape survival landscapes.

14.3 Entities, State Variables, and Scales

14.3.1 Spatial and Temporal Scales

Spatial Unit: Patch (3 m x 3 m resolution)

Temporal Unit: 5 minute time steps (tick)

14.3.2 Patch Variables

Variable Name	Definition
Suspended-particulate-matter SPM_t	Level of suspended material in patch (exponential difficulty increase between min and max observed in system) worse vision at max, best vision at min
Prey-Eaten-in-Patch	Counts how much prey are consumed in a patch
Prey-in-Patch $prey - in - patch_t$	The amount of prey currently in the patch
prey-alarmed?	Boolean variable set to true if any prey in the patch initiates fleeing behavior, triggering collective escape.

14.3.3 Prey: Agent Variables

Variable Name	Definition
vision	Radius (in patches) within which the prey can detect predators. Decreases with high SPM.
fleeing?	Boolean indicator of whether the prey is actively escaping from a predator.
energy	Energy levels of an agent
swimming speed	The speed at which a prey agent is moving
max-speed	Maximum normal swim speed (not in a flee state).
max-flee	Maximum achievable fleeing speed, calculated as a function of max-speed, swimming-speed, size, and max-rate-of-speed-change.
reaction-time	Time delay in response to predator presence. Increases with smaller size, higher SPM, larger/faster predators, and lower flee-ability.
size	Body length (size) of the prey; influences visibility and escape ability.
age	Agent's age class; mid-range age groups tend to have better escape success (a normal distribution across age classes).
flee-ability	Learning metric based on age; moderate-aged fish have best reflexes and performance.

14.3.4 Predator: Agent Variables

Variable Name	Definition
vision	Radius (in patches) within which predator can visually detect prey. Decreases with high SPM or low prey size.
swimming-speed	The speed at which a predator agent is moving
max-speed	Maximum normal swim speed (not in a burst state).

Variable Name	Definition
bursting?	Boolean indicating whether the predator is currently in a high-speed chase or attack behavior.
prey-in-vision	Agentset of prey within vision range
daytime-prey-eaten	Count of prey eaten that day
time-since-full	Time since last feeding (when will predator be hungry again?)
reaction-time	Delay between prey detection and initiation of chase. Influenced by predator size/age, prey density, prey speed, water clarity (SPM), and predation ability.
limit-daily-prey-allowance	Max prey allowed per day (predator becomes full)
gape-limit	Maximum size of prey the predator can successfully capture and ingest.
handling-effort	Time and energy cost required to subdue and consume a prey item. Depends on distance, prey size, and prey density. Used in patch and prey selection decisions.
size	Body length (size) of the prey; influences visibility and escape ability.
age	Age class of the predator agent. Predation success follows a normal distribution across age classes.
predation-ability	Agent's age class; mid-range age groups tend to have better predation success (a normal distribution across age classes).
prey-species-eaten	Species identifiers of consumed prey; enables diet tracking and multispecies modeling.
total-prey-eaten	Cumulative prey consumed across all days or scenarios.
digestion-time	Time it takes to digest prey

14.4 Process Overview and Scheduling

14.4.1 Prey Behavior (per agent)

1. **Fleeing Trigger**

Each prey evaluates whether a predator is in its visual cone or whether `prey-alarmed? = true` in the patch. If so, the prey calculates reaction time based on its own size, age, flee-ability, local SPM, and the predator's size and speed.

2. **Directional Escape**

If the reaction time threshold is met, the prey executes a directional fleeing behavior (`scare-left`, `scare-right`, `scare-down`, or `scare-up`), and sets the patch to `prey-alarmed? = true`.

3. **Schooling Response**

All other prey in the same patch automatically begin fleeing when `prey-alarmed? = true`, simulating social alarm propagation.

4. **Speed and Energy Update**

If the agent is fleeing, its speed increases toward `max-flee`, and energy is reduced based on movement cost. If not fleeing, the agent moves normally and may recover energy depending on its resting status.

14.4.2 Predator Behavior (per agent)

1. **Visual Detection**

The predator identifies prey within its in-cone vision, adjusted based on local SPM concentration, prey size, and predator size.

2. **Prey Filtering**

Prey that exceed the predator's gape limit or whose handling effort exceeds the predator's available energy are excluded.

3. **Prey Selection**

Among remaining prey, the predator evaluates candidates using an optimal foraging approach that considers size, handling effort, and net energy gain. One prey is selected.

4. **Pursuit Decision**

If a prey is selected, the predator enters a bursting state, increasing its speed toward `max-speed`. Reaction time is computed based on predator age, size, SPM, and prey proximity.

5. **Scare Prey**

The predator triggers fleeing in visible prey by calling the `scare-prey` procedure. If any prey flees, the patch is flagged as `prey-alarmed? = true`.

6. Consume Prey

If within striking range, the predator consumes the prey and updates variables: `daytime-prey-eaten`, `prey-species-eaten`, `total-prey-eaten`, and `Prey-Eaten-in-Patch`.

7. Digestion

If no prey are eaten, `time-since-eaten` increases. If the predator has reached its daily feeding limit, it enters a digestive phase. Digestion ends when both `time-since-eaten` and `time-since-full` exceed the digestion time

14.5 Design Concepts

Basic Principles: Simulates predator and prey interactions in a spatial estuarine system. Behaviors such as chasing, fleeing, and feeding are determined by individual traits, patch conditions, and environmental clarity due to suspended particulate matter.

Emergence: Predation hotspots develop in areas with higher prey density and better visibility. Prey movement and group alarm responses create shifting patterns of refuge and risk across the landscape.

Adaptation: Prey modify their direction and intensity of escape based on the predator's location and their own size, age, and escape ability. Predators filter out prey that are too large to consume and shift to resting behavior after reaching their feeding limit.

Objectives: Predators aim to select prey that provide the highest net energy gain while minimizing pursuit and handling costs. Prey aim to avoid detection or flee successfully using both individual detection and social alarm cues.

Learning: Learning is represented through Mid-aged agents generally perform better than very young or very old individuals.

Prediction:

Sensing: Agents rely on in-radius and in-cone vision to detect others. Detection range is influenced by water clarity, body size, and prey movement. Predators exclude prey that are too large to handle even if they fall within the visible field.

Interaction: Predators and prey interact through visual detection, spatial proximity, and energy transfer through feeding. Social interactions occur through alarm cue propagation at the patch level and school level.

Stochasticity: Prey selection, fleeing direction, and reaction timing include probabilistic variation. Randomness is used to reflect natural variability in movement and perception.

Collectives: When one prey detects a predator and flees, all prey in the same patch respond. This collective behavior improves predator detection and increases the likelihood of survival through schooling.

Observation: Key outputs include prey consumption, alarm responses, spatial predation distribution, and energy use of individual agents.

14.6 Initialization

Variable	Initialized Value	Justification
energy	100	Prey begin fully energized to allow immediate fleeing or normal movement
fleeing?	false	Prey are not actively escaping at the start
prey-alarmed?	false	Alarm cue inactive at start of simulation
daytime-prey-eaten	0	daily counter of prey eaten starts at zero
time-since-full	0	predator begins simulation hungry
gape-limit	function of size	Large predators have wider gape limits
handling-effort	calculated per encounter	Depends on prey traits and local patch conditions
reaction-time	size-, age-, and SPM-based	Prey and predator values dynamically computed each tick
flee-ability	based on age	Represents individual escape competence across age classes
predation-ability	based on age	Mid-age predators have fastest and most accurate prey responses
$swim_{max}$	$1.5 - 3 \frac{bodylengths}{sec}$	Typical value for sustained swimming speed in small pelagic fish (refer to Videler, 1993).

14.7 Submodels

14.7.1 Prey Detection Probability

Visual detection of prey by predators is scaled by local water clarity, prey size, and prey movement. The effective detection radius decreases as SPM increases and increases as prey size increases.

$$P_{detect} = \left(\frac{S_{prey}}{S_{max}} \right) \cdot e^{-SPM_t/\tau}$$

Where:

- P_{detect} is the probability of detecting a prey agent.
- S_{prey} is the size of the prey agent.
- S_{max} is the maximum size of any prey in the system.
- SPM_t is the suspended particulate matter concentration in the current patch.
- τ is a turbidity scaling constant.

Biological Justification: Larger prey are more visible, but high turbidity (SPM) reduces contrast and visual range. The exponential function reflects rapid degradation of visibility with increased turbidity.

14.7.2 Reaction Time

Reaction time determines how quickly an agent initiates a behavior in response to a threat (for prey) or opportunity (for predators). It is influenced by SPM, size, age, and ability.

$$RT_{agent} = \left(\frac{1}{F_{ability}} \cdot \frac{S_{opp}}{S_{agent}} \right) \cdot (1 + \alpha \cdot SPM_t)$$

Where:

- RT_{agent} is the reaction time of the prey or predator.
- $F_{ability}$ is flee-ability (for prey) or predation-ability (for predators).
- S_{opp} is the size of the opposing agent (predator or prey).

- S_{agent} is the size of the focal agent.
- SPM_t is the suspended particulate matter in the patch.
- α is the turbidity penalty factor.

Biological Justification: Smaller or younger fish tend to react slower. High turbidity delays response time. Age-related ability improves reaction time in middle-aged agents.

14.7.3 Alarm Response Propagation

$$P_{alarm} = 1 - e^{\frac{-n_{flee}}{\kappa}}$$

Where:

- P_{alarm} is the probability that non-detecting prey will flee.
- n_{flee} is the number of fleeing fish in the patch.
- κ is a sensitivity constant governing social responsiveness.

Biological Justification: Prey benefit from collective vigilance. The more individuals that flee in a patch, the more likely others will follow.

14.7.4 Fleeing Speed

The maximum speed a prey agent can reach when fleeing is scaled by body size, energy, and physical acceleration limits.

$$V_{flee} = V_{max} \cdot \left(\frac{E_{agent}}{100} \right) + \delta \cdot S_{agent}$$

Where:

- V_{flee} is the maximum fleeing speed.
- V_{max} is the prey's maximum speed.
- E_{agent} is the energy level of the prey agent.
- S_{agent} is the size of the prey.
- δ is a scaling factor for size-based speed increase.

Biological Justification: Larger prey generally swim faster, but energy limits how much of that speed can be used. Exhausted fish flee more slowly.

14.7.5 Catch Effort

Predators calculate the cost of pursuing and consuming prey using the prey's size and density, and the distance to the target.

$$H_{effort} = \gamma \cdot S_{prey} \cdot D_{patch} \cdot \left(1 + \frac{1}{\rho_{patch}}\right)$$

Where:

- H_{effort} is the handling effort for the predator.
- S_{prey} is the size of the prey.
- D_{patch} is the distance between predator and prey.
- ρ_{patch} is the prey density in the patch.
- γ is a handling cost coefficient.

Biological Justification: Prey that are farther away or more dispersed are harder to catch. Denser prey clusters reduce handling time by enabling rapid repeat captures.

14.7.6 Gape Filtering

Predators apply a binary filter before pursuit, excluding prey that exceed their maximum ingestible size.

$$P_{pursue} = \begin{cases} 1 & \text{if } S_{prey} \leq G_{pred} \\ 0 & \text{otherwise} \end{cases}$$

Where:

- P_{pursue} is the pursuit decision (1 = pursue, 0 = ignore).
- S_{prey} is the size of the prey.
- G_{pred} is the predator's gape limit.

Biological Justification: Predators cannot capture or ingest prey that are too large, so these individuals are ignored even if they are detected.

14.7.7 Energy Gain by Predators

$$E_{agent} = E_{agent} - E_{burst}$$

Where:

- E_{agent} is the current energy of the predator.
- E_{burst} is the energy cost per unit of bursting behavior.

Biological Justification: Bursting reduces available energy. Energy cost is a limiting factor for predation.

14.7.8 Energy Depletion by Prey

$$E_{agent} = E_{agent} - E_{flee}$$

Where:

- E_{agent} is the current energy of the prey.
- E_{flee} is the energy cost per unit of fleeing behavior.

Biological Justification: Fleeing reduces available energy and increases recovery time. Energy loss is a limiting factor for repeated escape attempts.

(no need to account for swimming energy in addition if you are using the migration functions)

Chapter 15

Model Building Tutorial: Staging & Schooling

15.0.1 Module Integration

- Brief explanation of how functions (e.g., staging, schooling) interact.
- Clarify temporal structure (e.g., tick-based sequence) and spatial scale.
- Describe coupling logic

15.0.2 Function Dependencies

- What variables are required as inputs for each function?
- What functions must be called before/after? (e.g., must calculate stress before checking staging triggers)
- Dependency table showing variable flow between submodels.

Function	Required Inputs	Output Variables	Dependent On
staging	E_agent, I_stress	E_agent, patch records	osmoregulation
schooling			

15.1 Implementation in NetLogo:

Programming details

example code

15.2 Implementation in R:

Programming details

example code

Chapter 16

Model Building Tutorial: Landward Migration & Selective Tidal Stream Transport

16.0.1 Module Integration

- Brief explanation of how functions (e.g., staging, schooling) interact.
- Clarify temporal structure (e.g., tick-based sequence) and spatial scale.
- Describe coupling logic

16.0.2 Function Dependencies

- What variables are required as inputs for each function?
- What functions must be called before/after? (e.g., must calculate stress before checking staging triggers)
- Dependency table showing variable flow between submodels.

Function	Required Inputs	Output Variables	Dependent On
osmoregulation	S_patch, S_agent, C	I_stress, C_new, E_osmo	environmental-sensing
staging	E_agent, I_stress	E_agent, patch records	osmoregulation

Function	Required Inputs	Output Variables	Dependent On
migration	E_agent, V_patch	Y_t, E_agent	staging, STST
STST	V_patch, V_agent	Y_t+1, E_agent	migration logic

16.1 Implementation in NetLogo:

Programming details

example code

16.2 Implementation in R:

Programming details

example code

Chapter 17

Complex Model Building Tutorial: One-Way Migration

17.0.1 Module Integration

- Brief explanation of how functions (e.g., staging, schooling) interact.
- Clarify temporal structure (e.g., tick-based sequence) and spatial scale.
- Describe coupling logic

17.0.2 Function Dependencies

- What variables are required as inputs for each function?
- What functions must be called before/after? (e.g., must calculate stress before checking staging triggers)
- Dependency table showing variable flow between submodels.

17.1 Implementation in NetLogo:

Programming details

example code

17.2 Implementation in R:

Programming details

example code

Chapter 18

Modeling Toolkit

18.1 Learning Resources

- **NetLogo Library:** Brief note on structure (e.g., model categories), how you used it (e.g., behavior ideas, calibration).
- **NetLogo User Manual:** Link to official documentation.
- **NetLogo Modeling Commons:** Peer-contributed models, code sharing, and idea sourcing.
- **NetLogo Forum:** Where to get help or search issues.
- **Book References:**
 - Railsback & Grimm (2019) **Agent-Based and Individual-Based Modeling**
 - Grimm & Railsback (2005) **Individual-Based Modeling and Ecology**
 - Add others specific to modeling.

18.2 Best Practices

- **ODD Protocol (Overview, Design concepts, Details):** Follow for transparent and structured documentation of agent-based models. Ensures clarity across entities, processes, and assumptions.
- **Modular Design:** Structure behavioral functions into separate procedures (e.g., swimming, osmoregulation, staging) to support testing, integration, and reuse.
- **Version Control:** Use GitHub or other tools to track changes and link code to research outputs.

- **Model File structure:**
- **Journal References:**

18.3 Code Repositories & Examples

Project	Repository	Description
Penobscot River	GitHub link	Simulates agent-based landward & seaward migration using STST.
Martha's Vineyard	GitHub link	Chloride cell regulation and stress-energy tradeoffs.

References

Glossary

This glossary defines key terms used throughout this book. Terms are organized alphabetically for clarity.

A

18.3.1 Acclimated Salinity

The salinity level a fish has gradually adjusted to and can tolerate without stress.

18.3.2 Agent

A virtual organism (e.g., fish) in the agent-based model that makes decisions and moves through space.

18.3.3 Allometric

A biological scaling relationship where physiological processes such as metabolism, growth, or filtration capacity scale nonlinearly with body size and/or age rather than proportionally (linearly).

18.3.4 Anadromous

Fish that hatch in freshwater, migrate to the ocean to grow, and return to freshwater to spawn.

18.3.5 Base Energy

The minimum energy required for basic physiological maintenance under resting conditions.

B

18.3.6 Benthic

Related to the bottom of a water body, where some fish feed or rest.

18.3.7 Bioaccumulation

The gradual buildup of contaminants like mercury in a fish over time.

18.3.8 Biomagnification

Increase in contaminant concentration as it moves up the food chain from prey to predator.

18.3.9 Bottom Feeding

A feeding strategy where fish forage along the riverbed or estuary floor.

18.3.10 Broadcast Spawning

A reproductive strategy where males and females release sperm and eggs freely into the water column.

C

18.3.11 Chloride Cells

Specialized gill cells that regulate salt balance during transitions between fresh and saltwater.

18.3.12 Contamination

Presence of toxins or pollutants (e.g., mercury) in water or sediment.

18.3.13 Current Velocity

Speed and direction of water movement at a location.

D**18.3.14 Diadromous**

Fish that migrate between freshwater and saltwater as part of their life cycle.

18.3.15 Digestion

The physiological process by which ingested biomass is broken down and converted into usable energy.

18.3.16 Depth (Bathymetry)

Vertical distance from water surface to the bottom; influences habitat and hydrodynamics.

E**18.3.17 Ebb**

Outgoing tide when water flows toward the sea.

18.3.18 Environmental Stress

Physiological stress caused by environmental conditions such as salinity or temperature extremes.

F**18.3.19 Filter Feeding**

Feeding method where fish strain tiny particles (plankton, detritus, SPM) from water using gill structures.

18.3.20 Flood

Incoming tide when water flows inland.

18.3.21 Flood Limit

The maximum upstream reach of tidal influence.

H**18.3.22 Homing**

The ability of fish to return to their birthplace to spawn.

18.3.23 Homeostasis

Maintenance of stable internal conditions such as temperature and ion balance.

I**18.3.24 Iteroparous**

Species that can spawn multiple times across their lifespan.

L

18.3.25 Landward Migration

Movement from the ocean into estuaries or rivers, often for spawning.

18.3.26 Lipid Catabolism

The mobilization and breakdown of stored lipids to meet energetic demands when food intake is insufficient.

M

18.3.27 Mercury

A toxic metal that accumulates in aquatic systems and organisms.

18.3.28 Metabolism

The total energy expenditure required for physiological maintenance, swimming, digestion, osmoregulation, and other biological processes.

18.3.29 Methylmercury

A highly toxic, bioaccumulative form of mercury.

O

18.3.30 Optimal Temperature

Temperature range where fish perform best physiologically.

18.3.31 Osmoregulation

Physiological process of regulating internal salt and water balance.

18.3.32 Osmoregulatory Energy

Energy required to maintain ion balance under varying salinities.

P

18.3.33 Pairwise Spawning

A reproductive strategy requiring a male and female to occupy the same location for fertilization.

18.3.34 Patch

A discrete spatial cell in an agent-based model with its own environmental attributes.

18.3.35 Pelagic

Relating to open-water habitats away from the bottom.

S

18.3.36 Salinity

Concentration of dissolved salts in water.

18.3.37 Salinity Stress

Energy strain caused when salinity deviates from a fish's preferred range.

18.3.38 Salt Wedge

A dense layer of seawater intruding beneath freshwater in an estuary.

18.3.39 Seaward Migration

Movement from rivers or estuaries toward the ocean.

18.3.40 Selective Tidal Stream Transport (STST)

Behavior where fish exploit tidal currents to assist migration.

18.3.41 Semelparous

Species that spawn once and then die.

18.3.42 Spawning

Release of eggs and sperm for reproduction.

18.3.43 Staging

A temporary resting or holding phase in migration.

18.3.44 Suspended Particulate Matter (SPM)

Fine particles (sediment, organic matter) suspended in the water column.

18.3.45 Swimming Difficulty

Effort required to swim based on flow direction and strength.

18.3.46 Swimming Energy

Energy expenditure required for locomotion.

18.3.47 Swimming Speed (Velocity)

Rate of movement relative to water.

T**18.3.48 Temperature**

Water temperature; major driver of metabolism and behavior.

18.3.49 Temperature Stress

Stress caused when water is too warm or too cold for optimal performance.

18.3.50 Thermal Tolerance

Range of temperatures within which a fish can survive and function.

18.3.51 Thermoregulation

Behavioral regulation of temperature by moving into warmer or cooler habitats.

Bibliography

- (NOAA) National Oceanic {and} Atmospheric Administration. Sediment quality guidelines developed for the national status and trends program. URL https://rais.ornl.gov/documents/ECO_BENCH_NOAA.pdf. Published: NOAA Technical Memorandum NOS OMA 52.
- C. Gaudet, S. Lingard, P. Cureton, K. Keenleyside, S. Smith, and G. Raju. Canadian environmental quality guidelines for mercury. 80(1):1149–1159. ISSN 0049-6979, 1573-2932. doi: 10.1007/BF01189777. URL <http://link.springer.com/10.1007/BF01189777>.
- Volker Grimm, Uta Berger, Donald L. DeAngelis, J. Gary Polhill, Jarl Giske, and Steven F. Railsback. The ODD protocol: A review and first update. 221(23):2760–2768, a. ISSN 03043800. doi: 10.1016/j.ecolmodel.2010.08.019. URL <https://linkinghub.elsevier.com/retrieve/pii/S030438001000414X>.
- Volker Grimm, Steven F. Railsback, Christian E. Vincenot, Uta Berger, Cara Gallagher, Donald L. DeAngelis, Bruce Edmonds, Jiaqi Ge, Jarl Giske, Jürgen Groeneveld, Alice S.A. Johnston, Alexander Milles, Jacob Nabe-Nielsen, J. Gareth Polhill, Viktoriia Radchuk, Marie-Sophie Rohwäder, Richard A. Stillman, Jan C. Thiele, and Daniel Ayllón. The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. 23(2):7, b. ISSN 1460-7425. doi: 10.18564/jasss.4259. URL <http://jasss.soc.surrey.ac.uk/23/2/7.html>.
- Volker Grimm, Eloy Revilla, Uta Berger, Florian Jeltsch, Wolf M. Mooij, Steven F. Railsback, Hans-Hermann Thulke, Jacob Weiner, Thorsten Wiegand, and Donald L. DeAngelis. Pattern-oriented modeling of agent-based complex systems: Lessons from ecology. 310(5750):987–991, c. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1116681. URL <https://www.science.org/doi/10.1126/science.1116681>.
- Raphael A. Lavoie, Timothy D. Jardine, Matthew M. Chumchal, Karen A. Kidd, and Linda M. Campbell. Biomagnification of mercury in aquatic food webs: A worldwide meta-analysis. 47(23):13385–13394. ISSN 0013-936X, 1520-5851. doi: 10.1021/es403103t. URL <https://pubs.acs.org/doi/10.1021/es403103t>.

Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. 21(4):25–34. ISSN 0097-8930. doi: 10.1145/37402.37406. URL <https://dl.acm.org/doi/10.1145/37402.37406>.