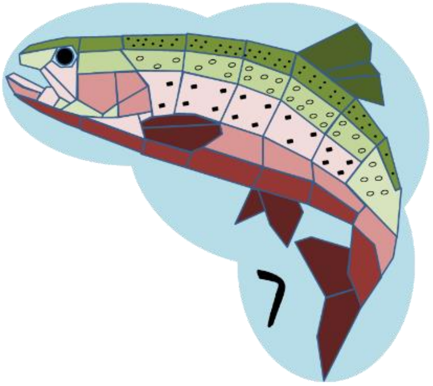
InSTREAM 7 User Manual: Model Description, Software Guide, and Application Guide



Prepared by:

Steven F. Railsback
Lang Railsback & Associates
Arcata, CA

Bret C. Harvey
Pacific Southwest Research Station
USDA Forest Service
Arcata, CA

Daniel Ayllón
Department of Biodiversity, Ecology and Evolution
Complutense University of Madrid, Spain

Last updated: July 7, 2023

This document is in preparation for publication as a General Technical Report of the USDA Forest Service, Pacific Southwest Research Station.

Revision Log

Date	**Section**	**Revision**	**Author**
8/5/2022	All	Fork document from USFS publication version. Set Normal style to single spacing. Add this revision log.	SFR
8/5/2022	17	Add troubleshooting item to guide: no fish spawn.	SFR
8/8/2022	12.1, 12.2, 15, 16.6	Emphasize use of standard output files for BehaviorSpace experiments instead of updating special variables for BehaviorSpace output.	SFR
8/8/2022	9.37	Add mention of other Brown Trout development rate studies cited by Réalis-Doyelle et al. (2016) and that they differ substantially.	SFR
8/8/2022	9.32, 9.33	Add citations for Brown Trout egg survival data.	SFR
11/3/2022	All	Check and correct date format codes (m/d/yyyy should be M/d/yyyy).	SFR
7/6/2023	9.18.1	Add reference to Källo et al. on length effect on terrestrial predation.	SFR
7/6/2023	9.22	Replace Fig. 31: temperature panel was duplicated and irradiance was missing.	SFR

Abstract

InSTREAM 7 is a major revision of the individual-based stream trout model we have developed and used for over 20 years. This user manual provides background information on inSTREAM and its purposes, a complete description of the model formulation, a software guide, and guidance on applying the model. InSTREAM is a simulation model designed to support river management decisions; it predicts how stream trout populations respond to habitat alteration including altered flow, temperature, and turbidity regimes and changes in channel morphology. The model represents individual trout, with population responses emerging from how individuals are affected by their habitat and by each other (especially, via competition for food). Key individual behaviors include habitat and activity selection (deciding whether to feed or hide, and selecting the best available location), feeding and growth, mortality, and spawning. Trout growth depends on prey availability and hydraulic conditions. Mortality risks due to terrestrial predators, piscivorous fish, and extreme conditions are functions of habitat and trout variables. InSTREAM 7 differs from previous versions in explicitly representing the daily light cycle via four time steps (dawn, day, dusk, night) and how light affects feeding, predation risk, and behavior; and representing activity selection as an additional adaptive behavior. InSTREAM 7 is newly implemented in the NetLogo software platform, which makes installation, use, and customization of the model much easier.

Keywords: Individual-based model, instream flow, population model, water temperature, Salmonidae, stream, trout, turbidity.

Summary

This user manual documents inSTREAM 7, a major update of the individual-based stream trout model we have developed and used for over 20 years. InSTREAM is a simulation model designed to understand and predict how stream and river salmonid populations respond to habitat alteration, especially altered flow, temperature, and turbidity regimes. This model has been used at sites ranging from headwaters to large regulated rivers for a variety of research and management questions. It is a complex model with many inputs and parameters, but because InSTREAM directly predicts the cumulative effect of multiple variables on fish populations, it does not require the kinds of untested assumptions and guesswork needed to base decisions on simpler models.

InSTREAM 7 is a major revision, with new software, a complete update of all assumptions and parameter values, and several key formulation changes. InSTREAM 7 is unique in treating light explicitly as a factor controlling feeding success, predation risk, and therefore trout behavior; the daily light cycle is represented as four phases, for dawn, day, dusk, and night. At each phase, model trout adaptively decide whether to feed or hide, and where to do so, as a tradeoff between growth and avoidance of predation risk.

Part I of the user manual provides background information and guidance on determining whether InSTREAM is a good tool for a particular situation.

Part II describes InSTREAM 7's formulation: its detailed assumptions and parameters and the information they were based upon. The model represents individual trout, with population responses emerging from how individuals are affected by their habitat and by each other (especially, via competition for food). Habitat is represented as microhabitat cells within one or several stream reaches. Key individual behaviors are habitat and activity selection (deciding when and where to feed), feeding and growth, mortality, and spawning. Fish growth depends on prey availability and hydraulic conditions. Mortality risks from terrestrial predators, piscivorous fish, and extreme conditions are functions of habitat and fish variables. Spawning and redd incubation are modeled simply while still representing effects of flow and temperature on reproductive success. Because it represents the full life cycle, InSTREAM is typically used to simulate populations over one or more decades.

Part III is a detailed guide to the model's software: how to install and run the model, prepare its input files, and control and interpret its output. InSTREAM 7 is newly implemented in NetLogo, a free, popular, and well-documented platform specifically for individual-based models. NetLogo provides a high-level programming language, graphical interfaces, and a powerful tool for automating simulation experiments.

Part IV provides guidance on applying inSTREAM 7 to new sites and management questions. It addresses preparation of site input, including field methods and development of input files, calibration, the design and interpretation of simulation experiments, and how inSTREAM has been and can be validated.

Contents

Figures

Tables

Part I: Introduction and Background

1 Introduction to InSTREAM 7

1.1 Document purpose and overview

This document provides a comprehensive user guide to version 7 of inSTREAM, the individual-based Stream Trout Research and Assessment Model. Version 7, first released in 2019, is a major revision of inSTREAM. Here we specifically describe inSTREAM version 7.3, released in 2021, but this document will generally apply to future releases of inSTREAM 7.

The user guide has 27 sections, divided into four parts:

- Part I (sects. 1 and 2) provides an overview of the guide and background information on inSTREAM, and establishes terminology and conventions used throughout the document.
- Part II (sects. 3 through 9) provides a complete description of the model: its assumptions, equations and algorithms, and parameters. The model description is linked to the model software via footnotes that describe where in the software each part of the model is implemented. Sects. 3 through 6 provide sufficient understanding of the model for most users, while sects. 7 through 9 provide complete details and justification of inSTREAM's many submodels.
- Part III (sects. 10 through 17) is a guide to inSTREAM 7's software, with information how to install and execute the model, prepare input files, control output, set up automated simulation experiments, and make common software modifications.
- Part IV (sects. 18 through 27) provides guidance on applying inSTREAM, from selecting study sites and collecting field data, through model calibration and the design and interpretation of simulation experiments for particular kinds of questions. Sect. 27 discusses how inSTREAM has been, and can be, validated.

The inSTREAM 7 software and additional supporting materials are distributed via the web site for individual-based ecological modeling at Humboldt State University: https://ecomodel.humboldt.edu. We encourage users and potential users of inSTREAM 7 to periodically check that site for updates: improvements and fixes to inSTREAM 7, its software, and this documentation will be posted there.

1.2 Why inSTREAM and individual-based modeling?

The development of inSTREAM (and its predecessor models: Jager et al. 1993; Van Winkle et al. 1996) was motivated by frustration with conventional assessment approaches that consider flow effects only through the concept of "habitat suitability" and temperature effects only by whether high-temperature thresholds are exceeded. Habitat suitability and temperature thresholds are extremely simplistic approaches that ignore most of what we know about trout physiology and ecology. Especially, inSTREAM is designed to overcome these specific limitations of conventional assessment methods:

- They are "static", not explicitly considering variation in conditions over time.

- They cannot make testable predictions of population responses to management, which limits their value in decision-making and makes the scientific cycle of model testing and improvement impossible.
- Empirically observed habitat “suitability” varies with, e.g., fish size, temperature, turbidity, season, time of day; is difficult to quantify reliably; and the availability of “suitable” habitat is not clearly a good predictor of population status (e.g., Garshelis 2000; Railsback et al. 2003). Observed habitat suitability relations are especially questionable for predicting responses to novel conditions.

Individual-based models (IBMs; Grimm and Railsback 2005) offer the potential to overcome these limitations by capturing the key mechanisms through which populations respond to time-varying habitat conditions and management. By representing the key habitat variables that affect individuals and the mechanisms through which those variables affect individual fitness (growth, survival, reproduction), an IBM can predict population responses by aggregating the fates of simulated individuals over time. A key feature of IBMs is representing adaptive behavior: how individual animals trade off the conflicting demands of growth, survival, and reproduction through behaviors such as selecting where to feed and when to hide instead of feeding. IBMs using this approach can predict population responses to realistically varying habitat conditions, making results directly applicable to management decisions and testable against field observations.

InSTREAM applies this approach to stream trout management (Figure 1). The model is driven by dynamic (time-varying) input for three main management variables: flow, temperature, and turbidity. These inputs control local (spatially variable) habitat variables that directly affect trout fitness, e.g., food availability, depth, velocity, availability of cover for feeding and escaping predators, and predation risk. The model trout then use adaptive behavior to determine, throughout the daily light cycle, whether to feed or hide and where to do so. Individual growth, survival, and spawning success therefore depend on habitat conditions and competition with other trout, mediated by behavior. Over simulated time, population responses to the inputs emerge from the fates of individual trout.

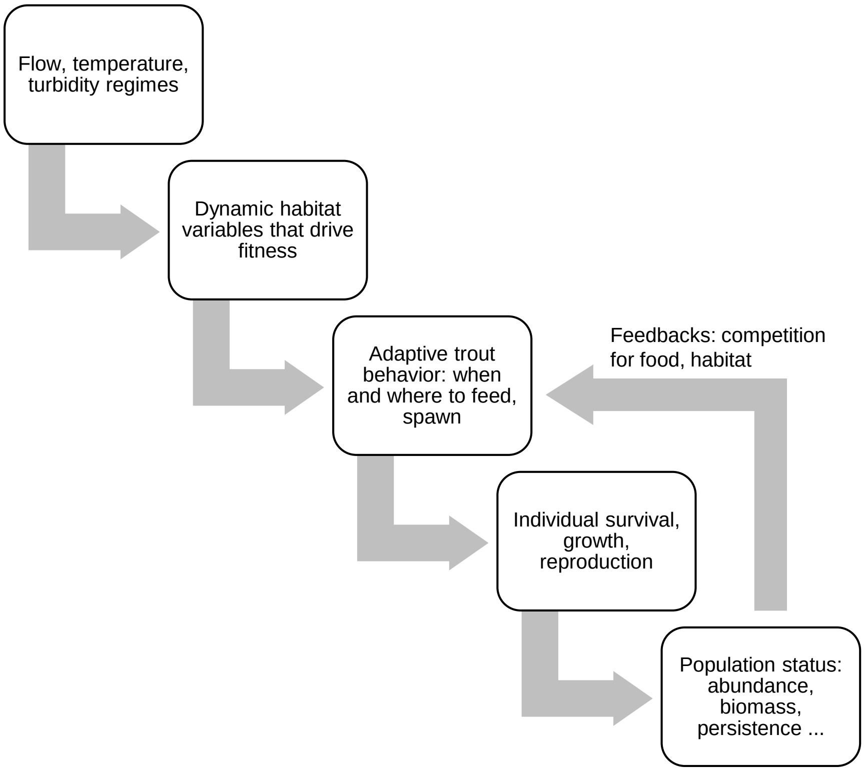

Flow, temperature, turbidity regimes
Dynamic habitat variables that drive fitness
Adaptive trout behavior: when and where to feed, spawn
Feedbacks: competition for food, habitat
Individual survival, growth, reproduction
Population status: abundance, biomass, persistence ...

Figure 1. Conceptual diagram of inSTREAM 7. From bottom to top: Predicted trout population characteristics of management significance are aggregated from of the fates of individuals: how many survive to what size and reproduce successfully. Individual fate is determined by habitat conditions but also by the adaptive behavior used to balance demands of growth, survival, and reproduction. Adaptive behavior is affected by feedbacks from the rest of the population: the options available to an individual and their benefits depend on what competing individuals do. Habitat conditions are driven by time-varying regimes of flow, temperature, and turbidity, as well as time of day.

1.3 Is inSTREAM the right tool?

Potential users of inSTREAM 7 must start by determining whether the model is an appropriate tool for their particular river management or research problems. In general, inSTREAM was designed for studies of how changes in habitat (channel morphology, flow, temperature, and turbidity) affect the long-term production and persistence of trout populations. InSTREAM has also proven useful for studies of how trout populations are affected by the biological processes represented in the model. Examples include examining interactions and competition among multiple trout species (or even among age classes of the same species) and effects of changes in food production or predation risk. Sect. 3.1 lists specific kinds of assessment and research problems that inSTREAM was designed to address.

InSTREAM is designed as much as a way to think about and understand river management questions as a way to predict the consequences of specific actions, because it includes the major mechanisms through which we know physical habitat, temperature, and biotic interactions affect trout individuals and populations. Applying inSTREAM to a management situation has benefits even if its results are not used directly in decision-making: it provides a checklist of important mechanisms to consider and defines what input and data are needed to evaluate those mechanisms. InSTREAM is also very useful as a field study planning and design tool, both for studies intended to support a decision process and for post-decision monitoring. Designing studies so that they provide input to inSTREAM or test the model's assumptions ensures that those studies will be directly useful for making and evaluating management decisions. Using inSTREAM encourages us to think explicitly about processes that drive trout populations—feeding and growth, predation risk, spawning success—in a way that directly supports management decisions.

No clear limits define the physical characteristics of sites (size, slope, geomorphology, etc.) where inSTREAM could be applied. However, we encourage users to understand the model's assumptions and submodels well enough to determine their appropriateness for specific applications. For any application, we recommend that users review model details and determine whether any processes should be turned off or altered.

InSTREAM will not always be the right model. It cannot be applied where sufficient site-specific input is unavailable or cannot be reasonably approximated. Not all study sites and problems will be compatible with inSTREAM's fundamental assumptions, and some involve processes that are not adequately represented in the model. Examples of sites or problems where inSTREAM may not be appropriate (without modification) include:

- Sites where non-salmonid species are significant competitors for food or habitat.
- Sites where water quality constituents other than temperature and turbidity have strong effects or are the management issues of interest. For example, standard versions of inSTREAM do not consider dissolved oxygen, or the effects of fine sediment on egg incubation.
- Study problems involving adaptive behaviors not in inSTREAM 7. For example, the standard versions do not represent how all life history characteristics adapt via behavior (e.g., adaptive selection of spawning age or size), and do not include evolution, so it is not useful for many problems of life history adaptation. (But note: The steelhead version of Railsback et al. 2014 provides an example of how life history adaptation can be added to inSTREAM, while Ayllón and colleagues, e.g., Ayllón et al. 2016, have modified inSTREAM to include trait evolution.)

InSTREAM also does not represent ice and its effects. Ice can cause direct mortality, alter or exclude habitat, reduce invertebrate food production via scouring, and provide protection from predation. InSTREAM excludes these processes because they are difficult to model; even the presence of ice is difficult to predict. Potential applications of inSTREAM to sites with ice may reasonably neglect ice effects, or represent them in the model in a simple way. Management actions that do not affect ice (e.g., changes in turbidity or summer flow) could be evaluated with inSTREAM by assuming ice effects are constant across alternatives. But inSTREAM would require significant modification to provide a mechanistic representation of how trout populations respond to management actions that strongly affect ice (e.g., altering winter flow or temperature).

The complexity of inSTREAM is also a concern in deciding whether to use it, especially in contentious situations where methods and results will be heavily scrutinized. There is a widespread belief that models such as inSTREAM that have many assumptions and parameters are inherently more uncertain and more subject to bias than simpler models. However, inSTREAM is not clearly more uncertain or subject to bias than traditional instream flow models (Railsback 1999; 2000; 2016). First, inSTREAM explicitly represents important processes neglected in models such as PHABSIM (Bovee et al. 1998); when left out of models, these processes are either ignored or must be dealt with ad hoc as decisions are made, adding uncertainty. Dealing with results for multiple life stages is a perfect example: PHABSIM produces separate results for spawning, juveniles, and adults, and there are no well-justified methods for combining these results into meaningful population-level results. InSTREAM directly predicts population status from what happens at every life stage. Second, habitat preference models such as PHABSIM depend almost entirely on preference curves, and these curves are essentially a large parameter set. Hence, these models also have many parameters. Third, this document defines and justifies each equation and parameter value, and almost all inSTREAM parameters can be independently evaluated. Finally, the robustness analyses described in Sect. 26.5 show that inSTREAM's results for decision-making—the ranking of management alternatives by the trout populations they provide—can be quite robust to parameter uncertainty.

Unfortunately, inSTREAM could not be designed as a generic model of stream fish; it is not readily applicable to species that differ from trout in fundamental ways. InSTREAM specifically assumes that individuals feed within a relatively small area for an entire time step (Sect. 4.2), so it cannot represent fish that forage over large areas in short time periods. While inSTREAM includes an alternative to drift feeding (Sect. 9.23), the search feeding submodel is very simple. InSTREAM's assumption that fish either feed or avoid predators over an entire time step is not useful for fish that use less digestible food than trout do: while inSTREAM assumes growth to be limited by the rate of food intake, the growth of grazers that consume relatively undigestible prey is typically limited by digestion time. Such fish can fill their gut relatively quickly but convert food to energy slowly; consequently, their behavior is likely less about feeding and more about avoiding predation while they digest. InSTREAM also neglects schooling as a risk reduction behavior, limiting its applicability to species that depend on that behavior.

1.4 History and evolution of inSTREAM; what is new in inSTREAM 7

InSTREAM 7 is the seventh major version in a family of individual-based salmonid models that have been in development since 1998 by the USDA Forest Service's Pacific Southwest Research Station (Redwood Sciences Laboratory), Lang Railsback & Associates, Humboldt State University, and other partners. Previous versions included:

- A single-species model specifically for our Little Jones Creek study site (Railsback and Harvey 2001);
- Addition of the capability to simulate multiple linked stream reaches and arbitrary numbers of trout species (Railsback et al. 2009);
- Two versions with sub-daily time steps that represent the difference between day and night and within-day flow fluctuations from hydropower operations (Railsback et al. 2005; Railsback and Sheppard 2014);
- Addition of the ability to drive habitat simulations with modern two-dimensional hydraulic models; and

- InSALMO, a version that represents freshwater life stages of salmon and Steelhead Trout (Railsback et al. 2013; 2014).

Railsback et al. (2009) provide additional background on the origins and early development of inSTREAM. The first version of inSTREAM was based in part on the trout IBM of Van Winkle et al. (1996, 1998), and inSTREAM 7 still contains some methods from that model. In a separate research program, Daniel Ayllón and colleagues at the Helmholtz Centre for Environmental Research-UFZ (Germany) and the Complutense University of Madrid (UCM, Spain) developed their own version ("inSTREAM-Gen") that includes evolution of life history traits. They apply inSTREAM-Gen to important research and management issues (e.g., Ayllón et al. 2016, 2018, 2019a, b, 2021).

InSTREAM 7 differs in many ways from previous versions, both in the model formulation and its implementation in software. Major changes from all previous versions include:

- There are by default four times steps per day, representing dawn, daytime, dusk, and night; previous versions used either one step per day or two steps representing daytime and night. Our simulation experiments indicate that this inclusion of night and crepuscular periods can affect model results in ways likely to be important to management decisions (Railsback et al. 2021).

- Light is now an important, explicit driver of model processes. The duration of each time step is determined by day length and how it varies with date and latitude. Light intensity (as irradiance) is calculated for each cell as a function of the time step's mean surface irradiance, turbidity, and cell depth; it affects both feeding and predation risk.

- The model now quantifies predation risk as the combined effects of several factors assumed to reduce risk, rather than as the effect of only the most prominent of those factors (sects. 9.18 and 9.19).

- InSTREAM 7 is programmed in the NetLogo modeling platform (Wilensky 1999); previous versions were programmed in Objective C and the Swarm simulation library, with separate graphical user interface software. The change to NetLogo makes the code much shorter, easier to understand, and easier to modify and maintain.

- Some spatial input is imported directly from geographical information systems (GIS) instead of requiring export to intermediate file formats.

2 Terminology and Conventions

2.1 Terminology

This section provides inSTREAM-specific definitions of terms used throughout the document. Much of the terminology is adopted from IBM terminology used by Railsback and Grimm (2019), and the ODD (Overview, Design concepts, Details; Grimm et al. 2006, 2010, 2020) protocol we use to describe inSTREAM. In this section only, terms defined here are set in bold typeface when used in defining other terms.

Action. An element in an IBM's **schedule**. An action is defined by a list of model **entities**, the **submodel** that the entities execute, and the order in which the entities execute the submodel.

Activity. One of three behaviors that model trout choose among: **hiding**, **drift feeding**, and **search feeding**.

Activity selection. The trout behavior of selecting an activity; however, activity selection is contingent on habitat selection and implemented in the **habitat and activity selection** submodel.

Behavior, **individual behavior**, **system behavior**. What a model trout or the model trout population actually does during a simulation, as a result of executing the **submodels**.

Cell. A model **entity** that is the basic unit of habitat in inSTREAM; habitat conditions vary among cells, but not within a cell. Cells are represented as irregular polygons.

Dawn. The **diel phase** defined as the morning period when the sun is between 6° below the horizon and 6° above the horizon.

Day. A 24-hour unit of time. Unless otherwise specified, a day is a calendar day, starting and ending at midnight.

Daytime. The part of a simulated day when the sun is above the horizon by more than 6° (the angle defining **dawn** and **dusk**).

Diel phase (or **light phase**, or simply **phase**). One time period (or one **time step**) defined by its light level; or the term used to describe the light level. The phases in inSTREAM 7 are **night**, **dawn**, **daytime**, and **dusk**.

Drift feeding. The activity that represents classic "sit and wait" feeding on drifting food (e.g., Hughes 1992).

Dusk. The **diel phase** defined as the evening period when the sun is descending from 6° above to 6° below the horizon.

Escape cover. Places where trout can temporarily avoid detection or capture by a predator when using the **drift** or **search feeding activity**. Along with **hiding cover** and **velocity shelter**, one of three habitat characteristics modeled separately in inSTREAM that fit under the general term "cover" but in fact provide different functions (Allouche 2002).

Eggs. Trout eggs are not represented explicitly in inSTREAM, but only via redd **state variables**. Therefore, when the model description mentions events that happen to eggs (e.g., *x*% of eggs die), it refers to events actually implemented as changes in the value of redd variables.

Entity. Following the ODD convention, something represented in the model as a discrete object with its own **state variables**. Example entities include individual trout, redds, and cells. The term "kind of entity" refers to entities of the same type that use the same submodels; "trout" is a kind of entity, while each simulated trout is an entity.

Fish. A synonym for **trout**.

Habitat and activity selection. The trout **behavior** of selecting which cell to occupy and which **activity** to use there, or the **submodel** that specifies how this decision is made.

Hiding, hiding cover, **hiding place**. Terms used in inSTREAM 7 explicitly in reference to a long-term (one time step or more) behavior of remaining concealed from predators instead of feeding (this behavior is also referred to as "concealment"). **Hiding** is an **activity** that precludes feeding and reduces predation risk. **Hiding cover** is habitat that trout occupy when using the hiding activity, where they are relatively safe from predation while not having to swim. For terrestrial predation, hiding cover is modeled as discrete **hiding places** that each can be occupied by at most one individual.

Initialization. The process of creating the model **entities** that are present at the start of a simulation and setting the values of their variables, before simulations start.

Initialization data. Data (GIS files, lookup tables, parameters, etc.) that are used in initialization.

Input data. Time-series input read and used to update conditions during simulations. Input data for inSTREAM 7 are time series of flow, temperature, and turbidity.

Light phase. See **diel phase**, above.

Model. In this document "model" refers to inSTREAM 7 in its entirety (e.g., "the model represents trout assemblages...").

Mortality source. A process (e.g., starvation, predation) that can cause trout or eggs to die. InSTREAM represents mortality sources as **survival submodels**.

Night. The **phase** when simulated light levels are lowest, representing the last part of one **day** and beginning of the next. Night begins when the sun exceeds 6° below the horizon and ends when the sun reaches 6° below the horizon the next day.

Observation, observer tools. The process of collecting data and information from the model; typical observations include graphical displays of habitat and fish over space and time and file output of summary statistics. Observer tools are software tools such as graphical user interfaces that make certain kinds of observation possible.

Observer. The model **entity** that controls overall execution (a NetLogo term we adopt). Global variables are treated as **state variables** of the observer; global variables and observer variables are synonymous. Procedures that control the model software at the top level, e.g., by initializing the model and executing its schedule, are executed by the observer, so global procedures and observer procedures are synonymous.

Parameter. A model variable meeting the conventions established in Sect. 2.2.2. Parameters are typically used as coefficients for the equations used in **submodels**. Contrast with **state variable** and **parameter value**.

Parameter value. A user-specified value for one of the model's parameters. Parameter values are one kind of input, but not to be confused with input data or initialization data.

Phase. See **diel phase**, above.

Procedure. A module of the model's software that encodes one submodel or part of a submodel. (This NetLogo term corresponds to software terms such as "method" or "subroutine".)

Reach. A habitat **entity** that represents a length of stream, and consists of a collection of **cells**. A reach typically corresponds to a field study site.

Redd. A model **entity** that represents a redd (nest of eggs) created by a spawning trout and the eggs in it.

Replicate. One of multiple models runs that represent the same **scenario** but use different pseudo-random number sequences that drive the model's stochastic processes (Sect. 6.9). Replicates reveal the magnitude of variation in results due only to the model's stochasticity.

Scenario. A single, complete set of model input, including **input data**, **initialization data**, and **parameter values**. Each scenario typically represents one particular set of environmental conditions or one management alternative.

Schedule. A description of the order in which events occur during a simulation: the schedule defines the **actions** and the rules for executing them. In an IBM's software, the schedule is the code which defines actions and controls the order in which they are executed.

Search feeding. The activity that represents feeding on stationary food by actively searching for it (e.g., Nakano et al. 1999).

State variable. A measure of the status of a model **entity** (a trout, a habitat cell, the population) that is described using a single number or symbol. State variables may be static (constant over time, e.g., cell habitat characteristics read from initialization data) or dynamic (updated over time by model calculations).

Submodel. A part of an IBM's formulation that models one **behavior** or process, including processes executed only during **initialization**. Dividing inSTREAM into submodels allows each process to be modeled, calibrated, and tested separately.

Superindividual. A trout entity that represents multiple real trout. The user can choose whether trout are represented as superindividuals from the time they emerge as new trout until they reach a specified size.

Survival submodel. A **submodel** of a **mortality source**; it calculates a fish's probability of surviving (or the survival probability of a redd's eggs) for one particular kind of mortality for one day.

Time step. A unit of model execution: one cycle through the model's **schedule**. Each time step represents one **diel phase**.

Time step length. The amount of time represented by a particular **time step**. The time step length varies from step to step.

Trout. The **entities** that represent individual trout after they have progressed from eggs into free-swimming life stages. Except where explicitly noted otherwise, the term refers to simulated, not real, fish.

Velocity shelter. Space within a cell that reduces the speed at which trout must swim while using the **drift feeding activity**. Velocity shelter is typically provided by obstacles to flow such as boulders.

2.2 Conventions

This subsection establishes conventions used throughout this document and the inSTREAM 7 software. These conventions are very important for reducing the risk of error (e.g., due to incorrect units in equations or parameter values) and for facilitating model modifications and interpretation of output.

2.2.1 Text formatting

In this document we use text formatting to distinguish several model elements. Model variables are set in *this variable font* (italics). Variables treated as parameters are set in ***this parameter font*** (bold italics). Names of procedures in the model's code and excerpts from the code are set in `this code font`.

2.2.2 Parameters and parameter types

Parameters are defined in ODD as equation coefficients, but it can be hard to define exactly which variables are and are not parameters. We define parameters as static variables with values that, in addition to not changing over time, do not differ among the entities (reaches, cells, trout of the same species) using them. Parameter values are set during model initialization. We define the following parameter types by which entities (as defined in Sect. 4.1) use the parameters:

- **Model parameters** (also called **observer parameters**) are used by the observer and possibly other model entities. All entities use the same values of these parameters.
- **Reach parameters** are used by habitat reaches and possibly by other entities (cells, trout, redds) in the reach. Each reach can have its own value of these parameters. Parameters used by cells are treated as reach parameters because all cells of a reach use the same values.
- **Trout parameters** are used by trout and redds. Each trout species has its own values for trout parameters.

Parameters used in the survival submodels (sects. 9.14 and 9.31) deserve clarification because many mortality sources depend on characteristics of both the fish (or redds) and their habitat. For simplicity, we use the convention that all parameters for survival submodels are trout parameters, with a few important exceptions. The two exceptions for trout survival are the two parameters driving the general intensity of predation by terrestrial animals (Sect. 9.18) and predation by fish (Sect. 9.19), which are reach parameters so that predation intensity can be varied among reaches. The exceptions for redd survival are parameters for redd scour (Sect. 9.35) that are also reach parameters because they depend on channel geometry and typically vary among reaches.

2.2.3 Units

The inSTREAM 7 formulation and software consistently use these measurement units.

Distance and length are in centimeters (cm), and, therefore, areas are in cm^2, volumes in cm^3, and velocities in cm per second (cm/s). There are three exceptions to this convention. Stream flow is in units of cubic meters per second (m^3/s) because cm^3/s is an unfamiliar and cumbersome unit. The habitat input that defines the size, location, and characteristics of cells are in GIS format (shapefiles) that use distances in meters (m; Sect. 7.2). The light units (defined below) do not follow this convention. However, all internal variables and outputs involving depth, velocity, area, or distance use length units of cm.

Weight is in grams (g).

Time is in days (d). For example, time step lengths typically have values such as 0.47 d. Keeping all time units in d simplifies survival calculations. (Flow and velocity variables are an exception.)

Temperature is in Centigrade (°C).

Energy in bioenergetic and growth calculations is in Joules (J).

Light intensity is quantified as irradiance, in its standard units of W/m^2 (equivalent to $J/m^2/s$).

Turbidity is in nephelometric turbidity units (NTU).

Survival variables and parameters all are the probability of surviving for one day, unless a different time period is specifically stated.

Fish lengths are fork lengths (cm).

Fish and prey (food) weight variables use wet weight (g).

2.2.4 Parameter and variable names

In both this model description document and the model software, we use long, descriptive names for variables and parameters instead of following the mathematics tradition of using concise symbols. This convention is important because the large number of variables would make concise symbols cumbersome and difficult to distinguish, and because it lets us use the same variable names in both the formulation and software.

Variable names follow the NetLogo convention of combining words with hyphens, and typically start with the kind of entity the variable or parameter belongs to, e.g., *cell-velocity-shelter-area*. However, in this document's equations we use the C programming language convention for variable names (e.g., *cellVelocityShelterArea*) to avoid confusion between hyphens and minus signs. Hence, *cell-velocity-shelter-area* and *cellVelocityShelterArea* are the same variable.

2.2.5 Mortality sources and survival submodels

A number of processes can kill fish or fish eggs in inSTREAM. We refer to these processes as mortality sources, but the model represents mortality via survival probabilities: the unit of mortality risk is the probability of surviving the mortality source for one day. (The term "mortality risk" is commonly used to mean the daily probability of dying, equal to one minus the survival probability.)

Modeling mortality as a survival probability simplifies computations and reduces the chances of error. The probability of surviving several mortality sources is calculated simply by multiplying their separate survival probabilities. Likewise, the probability of surviving one kind of mortality for *n* days (even when *n* is a fraction of one day) is calculated by raising the daily survival probability to the power *n*.

2.2.6 Dates and times

The software for inSTREAM 7 uses a sophisticated date-time library that allows input and output, and internal date variables, to use actual dates and times (e.g., in M/d/yyyy H:mm format). Leap days are automatically included, so must appear in time series inputs. However, the software does not consider daylight savings time so any sub-daily input must represent 24 hours in all days.

The conventions used by the date-time code number hours 0-23 and minutes and seconds 0-59.

2.2.7 Time step labeling in output

The time steps in inSTREAM 7 each represent a fraction of one day, with starting and ending times that vary among days. Model output is labeled with a date and time, and light phase, for the time step at which it was produced.

Time steps are labeled using conventions which, if not understood, can cause confusion in interpreting results. First, each model output (each line in an output file, or on the NetLogo interface) reports the model's state at the end of a time step. The date and time used to label the time step is when the time step *ends* (the time at which the step *started* is therefore the time reported for the previous time step). The phase is the light condition that occurred throughout the time step.

Therefore, the last of these output lines:

```
10/1/2000 17:12 Phase: day
10/1/2000 18:16 Phase: dusk
10/2/2000 05:44 Phase: night
```

was produced at the end of a time step that started at 18:16 on 1 October and ended at 5:44 on 2 October, during which it was night. The middle line was produced at the end of a step that represented 17:12 to 18:16 on 1 October, when it was dusk.

2.2.8 Fish ages and age classes

InSTREAM uses the convention that fish are age 0 when born and have their age incremented at the start of each January 1. (However, if a simulation starts on January 1 the birthday is skipped.) In producing output, inSTREAM summarizes results by age classes (Sect. 12.2). We typically use three age classes (but the software makes it very easy to change the number of classes; Sect. 14.1.5):

- Age 0—fish that have not yet reached their first January 1.
- Age 1—fish that have survived (as fish, not eggs) one January 1.

- Age 2+—any fish that have survived the January 1 of two or more years.

2.2.9 Habitat units and distances; adjacent cells

InSTREAM 7 uses a GIS-based format for depicting space. Cells are two-dimensional (in the horizontal plane) polygons that can have 3 or more sides. The X and Y coordinates used to define cell corners can be in any rectangular coordinate system, but follow conventions of the UTM system: coordinates are in meters, with X as the east-west dimension and X coordinates increasing from west to east, and Y increasing from south to north.

NetLogo has a built-in grid space and inSTREAM 7 uses many NetLogo commands that calculate distances in NetLogo's coordinate space. The built-in grid cells are called "patches" and NetLogo commands report distances in units of patch widths. InSTREAM uses an observer parameter ***world-resolution*** to convert NetLogo distances to cm: ***world-resolution*** is the number of cm represented by the width of a NetLogo patch, so NetLogo distances × ***world-resolution*** are actual distances in cm.

Cell velocities are modeled as magnitudes without any direction.

Important model processes depend on distances between cells (e.g., for finding all the cells within a fish's habitat selection range). The distance between two cells is calculated as the straight-line distance between their centroids. Centroid locations are determined from the GIS shapefile that represents the cells, including cells from multiple stream reaches when multiple reaches are simulated (Sect. 13.2).

The habitat and activity selection submodel (Sect. 9.13) assumes that even the smallest trout can always move at least to cells adjacent to their current cell. Cells adjacent to a starting cell are defined as those sharing at least one vertex (corner) or side with the starting cell[1].

2.2.10 Latitudes

InSTREAM 7 calculates the length and mean sunlight irradiance of each time step as a function of latitude and date. The model does not accommodate latitudes south of the equator, nor polar latitudes so far north that some summer days have no night and some winter days have no daytime. (Southern latitudes could be accommodated with a minor change to the day length submodel. Application to polar latitudes would require modification of the step timing submodel, e.g., by using an alternative definition of twilight, and revision of how time steps are defined; Sect. 9.6.)

2.2.11 Logistic functions and their parameters

The survival submodels and other submodels make extensive use of logistic functions, which are often useful for representing relations that vary nonlinearly between 0 and 1. The Y value of a logistic function ranges between zero to one as the X value varies over any range. In inSTREAM, logistic functions are defined via parameters that specify two points X_1 and X_9: the X values at which the Y value equals 0.1 and 0.9. These parameters typically have names that refer to the function, the X variable driving it, and whether they correspond to Y = 0.1 or Y = 0.9; for example, ***mort-fish-pred-D1*** and ***mort-fish-pred-D9*** are the parameters defining the

[1] The code identifying adjacent cells is in the procedure `build-cells`.

depths ("***D***") at which a logistic function relating survival of fish predation to depth has values of 0.1 and 0.9. (Note X variables typically quantify characteristics of the habitat or of individual fish.)

The equation for a logistic function is:

$$Y = \frac{exp(Z)}{1.0 + exp(Z)}$$

where, in inSTREAM, *Z* is calculated from the function's two parameters using the method in Sect. 16.4.1 of Railsback and Grimm (2019)[2]. For inSTREAM's conventions, Z is calculated as:

$$Z = exp\left(A + (B \times X)\right)$$

where B = -4.3944 / (X_1 – X_9) and A = -2.1972 – ($B \times X_1$).

Because of the exp() function in the equation for *Y*, logistic functions are particularly vulnerable to variable overflow/underflow errors, which occur when the computer tries to produce a floating-point number outside the range it can handle correctly. This range depends on the software platform and sometimes the hardware. To avoid these errors while providing the important ability to evaluate logistic functions over a wide range, inSTREAM 7 sets *Y* to 1.0 when *Z* is greater than 35 (corresponding to *Y* = 0.9999999999999993 in NetLogo 6.x) and sets *Y* to 0.0 when *Z* is less than -200 (corresponding to *Y* = 1.4E-87)[3].

[2] Logistic functions are initialized from parameters in procedure `create-logistic-with-table-and-params`. Their values are calculated in the procedure `evaluate-logistic`.

[3] These limits on *Z* are programmed in `evaluate-logistic`. Version 6.0.x of NetLogo can actually handle *Z* values between -745 and +709, regardless of the computer hardware. NetLogo raises an overflow error and halts if *Z* is above 709, but sets the result to 0.0 if *Z* is less than -745.

Part II: Model Description

3 ODD Element 1: Purpose and Patterns

This model description follows the ODD protocol (Grimm et al. 2006, 2010, 2020), a widely used format for describing IBMs. The first three sections (3 through 5) provide an overview of inSTREAM 7. Sect. 6 describes the model's conceptual basis using 11 standard design concepts. Sections 7 through 9 provide full details of inSTREAM's initialization, input data, and submodels.

3.1 Model purposes

The primary purpose of inSTREAM is to model, for understanding and prediction, the effects of river management actions on stream salmonid populations and assemblages of sympatric salmonid species. It is designed to predict the population consequences of specific management actions such as changes in flow regime, but also—often more importantly—as a framework for thinking about and studying the many ways that channel shape, physical habitat, flow, temperature, turbidity, and biological conditions affect salmonid populations.

The model's original purpose is as an instream flow assessment tool, for evaluating alternative flow and temperature regimes. This purpose specifically includes representing the complex interactions between flow and temperature that may be well-understood at the individual level but challenging to understand (e.g., Xu et al. 2010), much less predict, at the population level using field studies alone.

InSTREAM has also been used to evaluate other river management actions including changes in turbidity regimes, physical habitat modification or restoration, and alteration of biological conditions such as by introducing or eradicating species, modifying food production, or changing the kinds or intensity of predation. InSTREAM has also proven useful for assessing effects of large-scale processes that affect streams, such as climate change.

The model's purpose includes evaluating effects of river management on reproductive life stages, including spawning and egg incubation. Spawning and incubation are included in inSTREAM to allow such evaluation and to allow population simulations to be much longer than the life span of individual fish.

Because of its mechanistic nature, inSTREAM has also proven useful as a virtual ecosystem in which controlled experiments can address more fundamental ecological questions such as how adaptive behavior interacts with habitat and competition to drive population and community dynamics. InSTREAM can be considered a community and ecosystem model in the sense that it can represent multiple competing species of trout, and in the sense that it represents trophic interactions among trout populations, their prey, and their predators (Railsback and Harvey 2013).

3.2 Patterns used to design and evaluate the model

This section of the ODD protocol identifies characteristic patterns that have been observed in real systems and then used to design and evaluate the model. These patterns are "characteristic" in the sense that they are believed to be driven by the same mechanisms that we believe are important for the model's purposes: reproducing these patterns makes us more confident that the model is useful for its purposes. Previous versions of inSTREAM were evaluated using a variety of patterns that we also considered (but not explicitly, in a documented way) in designing inSTREAM 7. We used other patterns explicitly, to evaluate the theory used for habitat and activity selection behavior (Sect. 9.13). Unless noted otherwise, all the patterns apply to swimming life stages of stream salmonids in general, regardless of species and size.

3.2.1 Patterns driven by habitat selection

These patterns were documented and used by Railsback and Harvey (2002) to evaluate the theoretical approach used to model habitat selection in inSTREAM.

1. Hierarchical feeding: there is a consistent preference for specific feeding sites, dominant individuals displace others from the most preferred sites, and subdominant individuals occupy preferred sites when the dominant fish are removed from them.
2. Response to high flows: Trout move to stream margins during high flows and then return to previous feeding sites as flows recede.
3. Response to a larger competing species: The introduction of larger competitors causes juvenile trout to shift habitat, often toward higher velocities.
4. Response to predatory fish: When large trout are present, juvenile trout use shallower habitat on average than they do in the absence of large trout.
5. Variation in velocity preference with season: Seasonal variation in temperature and day length result in changes in habitat use, especially use of higher velocities at higher temperatures.
6. Changes in habitat use with food availability and energy reserves: Trout shift to higher-growth but riskier habitat when food availability or their energy reserves decline.

3.2.2 Patterns used to evaluate population dynamics

These patterns were documented and used by Railsback et al. (2002) to evaluate the population dynamics of an early version of inSTREAM.

7. A "self-thinning" relation: The log-log relation between mean individual weight and abundance, across age classes, is linear with a slope near -4/3.
8. A "critical survival time": There is a period of intense, density-dependent mortality in newly hatched trout.
9. Unstable population dynamics: Even in relatively simple and undisturbed streams, adult trout abundance can be highly variable from year to year.
10. Density dependence in growth: At least some populations exhibit strong density-dependence in growth of age 0 trout (e.g., Imre et al. 2010).
11. Effects of pools on large adults: Removing deep pool habitat reduces the abundance of large, old trout.

3.2.3 Patterns driven by diel habitat and activity selection

The third set of patterns were documented and used by Railsback et al. (2005) to evaluate the ability of inSTREAM version 3 to simulate how trout select both activity (feeding vs. hiding) and habitat during both daytime and night.

12. Individual variation in diel activity: Even within a population and age class, individuals differ in whether they feed during daytime or night.
13. Nocturnal feeding in slower velocities: On average, trout feeding at night use shallower and slower water than those feeding during the day.
14. Higher local densities at night: On average, trout feeding at night feed closer together than those feeding in daytime.
15. Less nocturnal feeding at high temperature: There is a negative relation between stream temperature and the fraction of feeding that occurs at night.
16. Effects of life history stage on activity pattern: Life stages (and individuals) with higher fitness motivation for energy acquisition, such as juveniles and adults recovering from spawning, on average feed more during the daytime than do other life stages.
17. Competition increases daytime feeding: More intense competition can cause individual trout to shift from feeding at night to daytime.
18. More daytime feeding when food availability or fish condition is low: As food availability or fish body condition decreases, feeding activity shifts from night to daytime.
19. Diel activity patterns depend on habitat: Habitat conditions such as flow and the availability of hiding cover can strongly affect the amount of feeding during daytime vs. night.

3.2.4 Patterns used to evaluate inSTREAM 7

Railsback et al. (2020) used seven patterns to evaluate the habitat and activity selection method of inSTREAM 7, along with several alternative methods. These patterns are similar to some of the preceding ones except for considering dawn and dusk as well as day and night.

20. More daytime feeding when food availability or fish condition is low.
21. More daytime and crepuscular feeding at higher temperatures.
22. Feedbacks of competition on circadian foraging patterns: the percentage of trout feeding in each light phase varies with trout density.
23. Foraging patterns are affected by circadian cycles in food availability: if drift food becomes more available in, e.g., crepuscular phases, then feeding activity in those phases increases.
24. Less daytime foraging under higher predation risk.
25. Foraging patterns are affected by physical habitat conditions (e.g., flow regime).
26. Foraging patterns vary with day length.

4 ODD Element 2: Entities, State Variables, and Scales

4.1 Entities and state variables

This element of ODD summarizes the model’s structure by listing the kinds of entities in it and their state variables, as defined in Sect. 2.1. We include only primary state variables and not secondary variables that can be calculated from the primary ones. InSTREAM 7 has five kinds of entities.

(Some previous versions of inSTREAM also included barriers, habitat entities representing features such as cascades, weirs, or culverts that limit the ability of fish to move up- and downstream. This capability allowed analysis of barrier effects on population persistence and distribution, e.g., Harvey and Railsback 2012, 2021. Barriers have not yet been implemented in inSTREAM 7; users interested in including barriers should check the inSTREAM web site for updates and contact us about implementing them.)

In this section, state variables are listed and defined very briefly. Detailed explanation of state variables—e.g., exactly what they represent, where their values come from, methods for their evaluation in the field, etc.—are in the descriptions of the submodels that use them (Sect. 9) and can be found by searching this document for the variable name or using its index.

4.1.1 The observer

The observer is a single entity that controls the global variables and global submodels[4]. Observer state variables (Table 1) are global variables that change over time. (Static observer variables, because they do not change over simulated time, are considered parameters and defined with the submodels that use them.)

Table 1: Observer state variables

Variable name	Variable type and units	Meaning
sim-time	Date-time (date and time variables include a calendar date and time of day, e.g., 28 March 2019 14:20)	The date and time at the end of the current time step.
prev-time	Date-time	The date and time at the start of the current time step (and therefore at the end of the previous time step).
step-length	Real number, d	The length (in fraction of a day) of the current time step: the difference between *sim-time* and *prev-time*.
day-length	Real number, d	The length of daytime (when the sun is above the horizon) for the current day; this is *not* the length of the day phase.
twilight-length	Real number, d	The length of one of the two civil twilight periods (dawn or dusk; when the sun is between 0 and 6° below the horizon); *not* the length of the dawn or dusk phases.
light-phase	Text equal to “night”, “dawn”, “day”, or “dusk”	The diel phase of the current time step.
daily-step-length-list	List of real numbers, d	A list of the lengths of the previous time steps in the 1.0 d period that ends at *sim-time*.

[4] Observer variables are defined in the `globals` statement at the start of the code.

Variable name	Variable type and units	Meaning
sunlight-irradiance	Real number, W/m^2	Mean sunlight irradiance at the water surface for the time step.

4.1.2 Reaches

A reach is a habitat entity that represents a continuous section of stream or river; the habitat within a reach is represented as a collection of cells (Sect. 4.1.3)[5]. InSTREAM 7 can represent one or more reaches. Reach state variables (Table 2) therefore represent intermediate-scale habitat conditions that vary among reaches but not among the cells within a reach. Static reach variables are treated as parameters and defined with the submodels that use them.

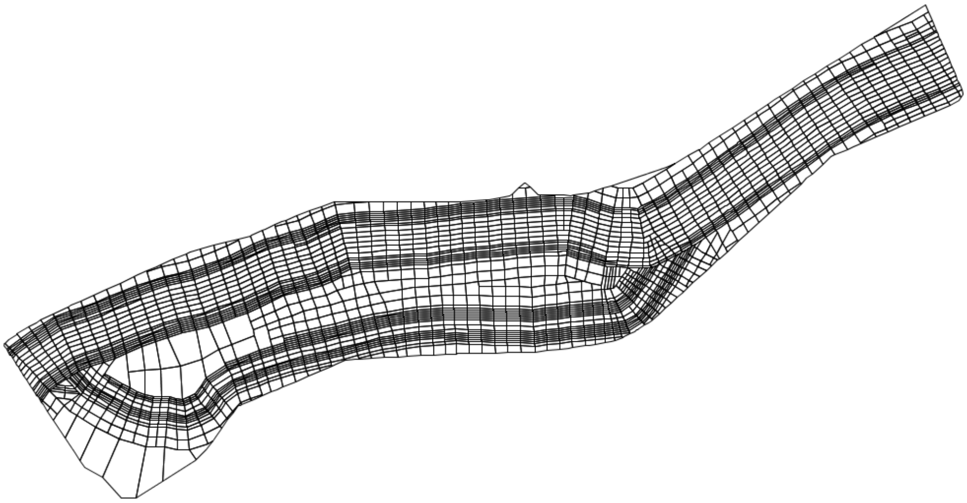


Figure 2: A reach and its cells.

Table 2: Reach state variables

Variable name	Variable type and units	Meaning
Flow	Real; m^3/s	The current stream flow.
Temperature	Real; °C	The current temperature.
Turbidity	Real; NTU	The current turbidity.
pisciv-trout-density	Real; trout/cm^2	The density of trout large enough to prey on other trout (Sect. 9.19.5).

[5] Reach variables are defined in the code statement `reaches-own`, near the top of the code.

4.1.3 Cells

Cells are the lowest-level habitat entities; habitat variation within a cell is not represented except via cell variables such as the fraction of the cell providing velocity shelter[6]. Each cell represents a two-dimensional polygon in the horizontal plane. Most habitat characteristics are represented as cell state variables (Table 3). Because they vary in value among the many cells, static cell variables are treated as state variables instead of as parameters.

Table 3: Cell state variables

Variable name	Variable type and units	Meaning
xcor, *ycor* (the software uses NetLogo's built-in coordinate variables)	Real, static; patch-widths.	The X and Y coordinates of the cell's centroid, converted to cm units by multiplying by ***world-resolution***.
cell-area	Real, static; cm^2	The cell's area.
cell-depth	Real, dynamic; cm	The cell's current depth; 0.0 if the cell is not submerged at the current flow.
cell-velocity	Real, dynamic; cm/s	The current water velocity; 0.0 when the cell is not submerged.
cell-light	Real, dynamic; W/m^2	The current irradiance at half of the cell's depth, or surface irradiance when the cell is not submerged.
cell-num-hiding-places	Integer, static; unitless	The number of places that each provide an isolated hiding place for one adult trout.
cell-escape-dist	Real, static; cm	A characteristic distance that a fish in the cell would have to move to enter escape cover.
cell-frac-vel-shelter	Real, static; unitless	The fraction of cell area providing velocity shelter for drift-feeding trout.
cell-frac-spawn	Real, static; unitless	The fraction of cell area providing substrate suitable for trout spawning.
cell-available-drift	Real, dynamic; g/d	The currently unconsumed drift food in the cell, expressed as a constant rate of food production.
cell-available-search	Real, dynamic; g/d	The currently unconsumed search food in the cell, expressed as a constant rate of food production.

[6] Cell variables are defined in the `cells-own` statement near the beginning of the code. *Cell-depth*, *cell-velocity*, and *cell-light* are coded as the patch variables *depth*, *velocity*, and *light* so they can be displayed in the graphical interface.

Variable name	Variable type and units	Meaning
cell-available-vel-shelter	Real, dynamic; cm^2	The currently unoccupied velocity shelter area.
cell-available-hiding-places	Integer, dynamic; unitless	The number of hiding places currently unoccupied by hiding trout.

4.1.4 Trout

InSTREAM represents the trout life cycle using only two kinds of entity, redds and trout. The trout entities represent individual trout from the time of emergence. (Here, "emergence" means the process of an egg developing into a juvenile trout that leaves the redd and enters the water column.) Trout characteristics are represented by the state variables in Table 4. Trout location is represented as the cell occupied by the trout: model trout do not have coordinates in model space but instead have a variable for which cell they occupy[7].

The model can represent an unlimited number of trout species, with species a state variable of trout. Species can differ from each other by using different trout parameter values but (without custom software modifications) all species use the same assumptions, equations, and submodels.

Model trout can optionally represent "superindividuals": agents that each represent a number of identical trout (see Sect. 7.9 of Grimm and Railsback 2005). Superindividuals are implemented for computational efficiency: they can reduce the number of model trout needed to represent the sometimes large numbers of newly emerged juveniles. Trout can be represented as superindividuals from the time they emerge (Sect. 9.38) until they reach a size at which they are turned into multiple individual trout (Sect. 9.39). The trout variable *trout-superind-rep* determines whether a simulated trout is a superindividual: this variable is the number of trout represented by the model entity and has a value of 1 for normal trout.

Table 4: Trout state variables

Variable name	Variable type and units	Meaning
trout-species	Text, static	The trout's species.
trout-sex	Text equal to "female" or "male"; static	The trout's gender.
trout-age	Integer, dynamic; years	Age, incremented by 1 each January 1 after the trout is created.
trout-length	Real, dynamic; cm	Length as fork length.
trout-weight	Real, dynamic; g	Wet weigh.

[7] Trout variables are defined in the `trout-own` statement near the beginning of the code. The location variable *trout-cell* does not actually exist in the code because NetLogo lets trout instead use the variable *patches-cell* of the patch they are on.

Variable name	Variable type and units	Meaning
trout-condition	Real, dynamic; unitless	Ratio (0-1) of the trout's weight to the weight of a healthy trout of the same length.
trout-max-speed	Real, dynamic; cm/s	Maximum sustainable swimming speed.
trout-activity	Text equal to "drift", "search", or "hide"; dynamic	The trout's selected activity on the current time step.
trout-cell	Cell; dynamic	The cell that the trout currently occupies.
trout-superind-rep	Integer; static	The number of trout represented by the model trout entity.
trout-spawned-this-season?	Boolean; dynamic	TRUE if the trout has spawned since the start of the current spawning season.
consumption-memory-list	List of real numbers, dynamic; g	A list of the food consumption the trout experienced during the previous time steps of the current 1.0-d period.
growth-memory-list	List of real numbers, dynamic; g/d	A list of the growth rates the trout experienced during the previous time steps of the current 1.0-d period.
survival-memory-list	List of real numbers, dynamic; survival probabilities per time step	A list of the survival probabilities experienced during the previous time steps of the current day.

4.1.5 Redds

Redd entities each represent a nest of eggs produced when a female trout spawns[8]. InSTREAM models the egg and alevin life stages of trout simply as state variables of redds (Table 5). Some important characteristics of redds (e.g., the depth they are buried; their area) are assumed not to vary among redds of the same species, and so are treated as parameters instead of state variables and defined with the relevant submodels.

[8] Redd variables are defined in the `redds-own` statement near the beginning of the code. Redds, like trout, can also use the variables of the patch they are on.

Table 5: Redd state variables

Variable name	Variable type and units	Meaning
redd-species	Text, static	The redd's species.
redd-num-eggs	Integer, dynamic; unitless	The number of viable eggs in the redd.
redd-frac-developed	Real, dynamic; unitless	The fraction (0-1) that the redd is of being fully developed; the redd starts turning eggs into new trout when this variable reaches 1.0.
redd-emerge-days	Integer, dynamic; d	The number of days that new trout have been emerging from the redd.

4.2 Model scales

4.2.1 Spatial resolution and extent

The spatial resolution of inSTREAM 7 refers to cell size and shape, and spatial extent refers to how much stream area is represented. User input entirely determines these critical characteristics of the model, so users must understand how cell size and spatial extent affect simulations. Here we simply describe typical spatial scales and identify potential effects of spatial resolution on model results. We provide specific guidance on the number and size of reaches to use in Sect. 19, and on selecting cell sizes and shapes in Sect. 22.

InSTREAM is designed to represent a wide range of spatial extents; our applications have ranged from modeling relatively short reaches of streams < 1 m across to representing many kilometers of rivers tens of meters wide. While the time needed to complete simulations increases with the amount of habitat included, the probability of extinction may become an issue for simulations of limited spatial extent.

Cell sizes also have lower limits, because small cells can cause simulation artifacts. The main factor we consider for cell size is that cells must be at least as large as the feeding territory of an adult trout (or a superindividual representing multiple small trout; Sect. 4.1.4); otherwise, the model's method of representing food availability and competition (Sect. 9.11.2) will cause cells to provide low fitness only as an artifact of their small size. Stream habitat modelers sometimes assume that finer resolution is always better because of the need to capture hydraulic features such as velocity shelters behind rocks; this is not the case with inSTREAM. Fine-scale hydraulic features such as velocity shelter are instead represented as variables of the cells, so they need not be represented as differences among small cells.

While cells can be too small, there is no clear upper limit on cell size except for the need to capture the site's variation in habitat. The food availability and competition methods potentially can introduce an artifact of large cell sizes: aggregating space into fewer but larger cells can increase the number of large trout that the space can support while decreasing the area with food enough for small, but not large, trout. (A small cell may have enough food for only one large trout, with enough left over for 1-2 juvenile trout. A cell 4 times larger may support 5-6 large trout and still only 1-2 juveniles.) One experiment varying cell size in an earlier version of inSTREAM (Butcher and Parrish 2006) indicated that this artifact was not strong.

A second potential artifact of cell size concerns the distance over which small trout can select habitat. The habitat and activity selection submodel (Sect. 9.13) assumes that trout evaluate and potentially move to cells within a radius that increases with trout length, but trout can always evaluate adjacent cells. Using fewer, larger cells means that adjacent cells can be bigger, effectively increasing the distance that small trout can move; on the other hand, using smaller cells tends to give small trout more cells to choose from.

Computer resources are another potential consideration for cell size; memory use and execution time increase with the number of cells, so using large numbers of unnecessarily small cells can be a computational burden.

Considering these effects of cell size, we recommend use of cells as small as—but not smaller than—necessary to capture important gradients in habitat characteristics like depth, velocity, and velocity shelter. Cells smaller than ~1 m^2 should be avoided, and we recommend breaking up very large cells (e.g., making up >50% of channel width) into several smaller cells. Sharp local gradients in velocity that provide velocity shelter can generally be represented via the cell's variable *cell-frac-vel-shelter,* not as separate cells.

4.2.2 Temporal resolution and extent

Temporal resolution and extent refer to time step lengths and how much total time is represented in a model run. User input partially determines temporal resolution and extent, so users must understand the range of possibilities and their consequences.

4.2.2.1 Time steps

The use of four time steps per day is a major change in inSTREAM 7; previous versions represented only one daytime step per day, or two representing daytime and night. InSTREAM 7 has separate time steps representing the four diel phases (dawn, daytime, dusk, and night) defined in Sect. 2.1. (Alternative versions of inSTREAM 7 that let time steps also be triggered by events such as within-day changes in flow or temperature are available.)

The step timing submodel (Sect. 9.6) calculates the starting time of each time step as a function of date and site latitude. Therefore, the time steps change each simulated day.

4.2.2.2 Model run duration

The observer parameters ***start-date*** and ***end-date*** specify the temporal extent of a model run. Simulations start with a time step representing the time from midnight to dawn of the day specified by ***start-date***, and end with the first time step (typically night) that ends after the day specified by ***end-date***.

An appropriate duration of an inSTREAM run depends on the purpose of the model application. For assessment of flow, temperature, or turbidity regimes, runs should be long enough to include key characteristics of the alternative regimes, such as differences in seasonal means and in the frequency and magnitude of extreme events. We often use runs of 10-50 years for such purposes.

For analyses of population viability—how the risk of local extinction varies among input scenarios—longer runs are appropriate. We have used runs of, e.g., 100 years for these analyses, which often requires synthesizing some of the time series input data because historic data are rarely available for such long periods.

Sometimes simulation studies focus on particular life stages or seasons, such as assessments of how alternative channel restoration designs affect spawning success or juvenile survival and growth. For such short-term objectives, individual seasons or years can be simulated. If the model is re-initialized each run, then short simulations of different years (e.g., spawning through 1 October) are independent and can be treated as replicates for analysis.

5 ODD Element 3: Process Overview and Scheduling

This section describes the model's schedule, the sequence of actions executed each time step[9]. The schedule is also a summary of the entire model. For each action in the schedule, we describe which entities execute which submodel or submodels, and why actions are scheduled in the order they are. ODD element 7 (Sect. 9) fully describes the submodels.

Describing each action includes specifying the order in which the model entities execute the action: for example, when trout select habitat and deplete the resources in the cell they select, the order in which trout execute habitat selection can have a strong effect on results. However, many actions do not include interaction among entities so the order in which entities execute does not matter. If the execution order is not specified below, then execution order has no effect and the action uses NetLogo's default, which is to randomize the order in which entities execute each time the action is executed.

Update actions. These actions are executed first because they set the new habitat conditions that trout then respond to.

Time updates[10]. The observer updates time variables. At the first time step of a day, the day length (Sect. 9.5) and step timing (Sect. 9.6) submodels determine the start times of the new day's time steps. The simulation time submodel (Sect. 9.7) advances the value of *sim-time* to the next such time and updates the values of *light-phase* and *step-length* accordingly. If the date for the new value of *sim-time* is after the model parameter ***end-date***, the simulation stops.

Habitat updates[11]. The observer updates habitat variables, using the habitat update submodel (Sect. 9.8). Reach-specific values of flow, temperature, and turbidity for the current time are read from input data. The depth and velocity of each cell are updated using the hydraulics submodel (Sect. 9.3). The surface light irradiance *sunlight-irradiance* is updated using the surface light submodel (Sect. 9.9). The light in each cell is updated using the cell light submodel (Sect. 9.10). Depletable cell resources are reset to undepleted values using the cell resource depletion submodel (Sect. 9.11).

Trout updates[12]. Each trout updates the memory list variables it uses in the habitat and activity selection submodel. This update is defined in the trout memory submodel (Sect. 9.12). If the current time step ends in a new calendar year (the year of *sim-time* is greater than the year of *prev-time*), then the age of all trout (*trout-age*) is incremented by 1 year.

[9] Following NetLogo convention, the schedule is programmed in the `go` procedure.

[10] The time updates action is programmed in the procedure `update-time-and-habitat`.

[11] The habitat updates action is programmed in `update-habitat`.

[12] The trout updates action is programmed in `update-trout` and `update-trout-memories`.

Trout habitat and activity selection, survival, and growth action. Each trout executes three submodels that represent their adaptation to current conditions and the resulting survival and growth. Each trout executes the three actions before the next trout executes. Trout execute this action in descending order of length (longest to shortest) to represent the size-based dominance hierarchy commonly observed in stream salmonids (e.g., Hughes 1992; Young 2003). When multiple trout have the same length, as when superindividuals are first separated into individual trout (Sect. 9.39), their order is randomized. The three submodels in this action are:

Habitat and activity selection. The trout decides whether to feed or hide, and which cell to occupy (Sect. 9.13). This submodel is first because trout use it to adapt to new habitat conditions.

Survival. The trout determines its probability of surviving six mortality sources, and then determines whether it survives each by using a random Bernoulli trial (which reports true or false, with the probability of true equaling the survival probability weighted by the length of the time step). Survival is scheduled before the growth action so that survival probabilities are calculated using the same fish variable values used for habitat and activity selection. (This scheduling means that growth on the current time step does not affect survival of starvation and disease.) The trout survival submodel (Sect. 9.14) details the overall action, but each mortality source has its own submodel. The order in which the survival of each mortality sources is executed could affect how many trout die of which kind of mortality, but this effect is expected to be small and the order is essential arbitrary. Mortality sources are evaluated in the following order:

High temperature acute stress (Sect. 9.15),

Stranding (Sect. 9.16),

Starvation and disease due to low condition (Sect. 9.17),

Predation by terrestrial animals (Sect. 9.18), and

Predation by fish (Sect. 9.19).

Growth. The trout updates its length and weight (Sect. 9.20).

Trout spawning action. This action is only executed once per day, on the first time step in daytime. All female trout use the spawning readiness submodel (Sect. 9.27) to determine whether they are ready to spawn, using criteria that consider habitat conditions, the date, and trout size, age, and condition. If a trout is ready to spawn, it does so by creating a new redd, possibly causing superimposition mortality on any other redds in the same cell, and incurring weight loss, according to the spawning submodel (Sect.9.28). This action is executed by female trout in descending order of length (longest to shortest) to represent a size-based dominance hierarchy in competition for spawning sites and mates.

Redd actions. Each redd action is executed by all redds before the next action is started. Because redd actions are scheduled after trout actions, new trout created by redds do not execute the trout actions on the time step they are created. This scheduling also means that new redds execute all the redd actions on the time step they are created.

Redd survival. The redd's value of *redd-num-eggs* is updated by determining how many of the redd's eggs survive each of several redd mortality sources, using the redd survival submodel (Sect. 9.31). The redd mortality sources, each with its own submodel, determine the probability of an egg surviving the mortality source for one day. For each mortality source, the value of *redd-num-eggs* is updated by setting it to a random draw from a binomial distribution (which models the integer number of true events in a certain number of independent trials, each with the same probability of being true) with *redd-num-eggs* trials and the probability of true equal to the survival probability weighted by time step length. The redd mortality sources are executed in the following order:

Low temperature (Sect. 9.32),

High temperature (Sect. 9.33),

Dewatering (Sect. 9.34), and

Scour (Sect. 9.35).

Redd development. The redd's value of *redd-frac-developed* is updated by an amount that depends on the temperature of redd's reach, as defined in the redd development submodel (Sect. 9.37). When *redd-frac-developed* reaches 1.0, the redd is considered fully developed, meaning that its surviving eggs are ready to emerge.

Redd emergence. This action is executed only once per day, on the first time step in daytime. Each fully developed redd converts some or all of its eggs into new trout, using the redd emergence submodel (Sect. 9.38). The number of eggs that "emerge" into new trout is a fraction of *redd-num-eggs* that increases over time, so that emergence from a redd is spread over several days in a humped distribution. The value of *redd-num-eggs* is reduced by the number of eggs that emerge, and the redd is removed from the model when *redd-num-eggs* reaches zero. For each emerging egg, a new trout is created and initialized using methods described in the redd emergence submodel. (Optionally, superindividuals that each represent multiple trout can be created, reducing the number of new trout entities.)

Output action. The output action is executed by the observer[13]. It is the last action in the schedule, so outputs reflect the state of the model at the end of the time step. The graphical interface is updated to display cells shaded by depth, velocity, or light; and the locations of trout and redds. Output files are updated. (The inSTREAM 7 software allows the user to schedule output file updates less frequently than every time step; Sect. 14.1.5.)

6 ODD Element 4: Design Concepts

This ODD element describes how the model implements 11 concepts of individual-based model design. These concepts are intended to capture key characteristics of individual-based models not easily described using conventional tools such as equations or flow charts.

6.1 Basic principles

InSTREAM was developed largely to supplant a firmly established basic principle, the idea that population and community effects of habitat alteration can be understood via habitat selection modeling. Specifically, inSTREAM was designed largely to overcome the limitations of the PHABSIM habitat selection model. The usefulness of habitat selection modeling has been challenged in river management ecology (Railsback et al. 2003; Railsback 2000, 2016) and in

[13] Output updates are programmed in the `update-output` procedure.

management ecology as a whole (e.g., Garshelis 2000; Johnson 2007). InSTREAM and its predecessors were designed to offer a different basic principle, the use of individual-based models that capture the most important mechanisms through which river management affects fish populations and assemblages.

InSTREAM 7 is largely a collection of existing models and modeling concepts assembled into a full-life-cycle population model. These models and concepts include: (1) conceptual and computation models of drift feeding (Sect. 9.22); (2) energy balance or "bioenergetic" models that relate growth, food intake, activity, and temperature (Sect. 9.21); and (3) redd development models (Sect. 9.37).

The approach used to model habitat and activity selection in inSTREAM (see Adaptation and Objectives, below) has been developed into a general approach to modeling adaptive tradeoff decisions in IBMs, called State- and Prediction-based Theory (Railsback and Harvey 2013, 2020). That approach is a modification of the general fitness maximization concept long used in behavioral ecology (e.g., Houston and McNamara 1999; Mangel and Clark 1986).

6.2 Emergence

All the primary outputs of inSTREAM (e.g., population abundance and biomass, persistence, age and size distributions) emerge from the model's mechanistic interactions among channel shape and physical habitat characteristics, food availability, the time-series inputs (flow, temperature, turbidity), and how fish physiology and behavior are modeled. Only a few of the model's intermediate outcomes are closely imposed by simple mechanisms and hence easily predicted; the most important of these is egg incubation time, which is determined from the temperature regime in a relatively simple way (Sect. 9.37).

6.3 Adaptation

Trout in inSTREAM have two adaptive behaviors: selecting habitat and activity each time step, and deciding when and where to spawn. Habitat and activity selection (Sect. 9.13) is modeled as direct objective-seeking: trout make this decision to maximize an objective function (outlined below) that is an explicit measure of their expected future fitness. In contrast, spawning behavior is modeled as indirect objective-seeking: inSTREAM uses algorithms and parameters that cause trout to reproduce spawning behavior of real trout instead of representing how the behavior affects expected fitness.

6.4 Objectives

The habitat and activity selection submodel (Sect. 9.13) assumes that trout select the combination of habitat cell and activity (whether to hide or feed, and which of the two feeding strategies to use) that maximizes an objective function, typically called a fitness measure in ecology. The fitness measure explicitly represents a trout's expected probability of surviving all kinds of mortality, including starvation, over a sliding time horizon: the next *n* days into the future. The time horizon is a trout parameter (***trout-fitness-horizon***) with a standard value of 60 days, which forces the trout to balance predation risk against the need to feed: starvation becomes a high risk over 60 days if a trout chooses only to hide. The fitness measure considers not just the habitat and activity chosen on the current time step but the combination of activities chosen over a full day.

6.5 Learning
InSTREAM 7 does not represent learning. (The decisions made by trout depend on their previous experience but trout do not change *how* they make decisions as a consequence of their experience.)

6.6 Prediction
The habitat and activity selection submodel assumes trout base their decisions on explicit predictions of future habitat and competition conditions. The predictions are extremely simple: trout assume that conditions such as physical habitat, food availability, and predation risk over the fitness time horizon will be the same, over the time steps of a full day (dawn, day, dusk, night), as they have been over the 1.0-d period that ends at the end of the current time step. In other words, trout base their decisions on the prediction that conditions experienced on the current day will persist over the entire time horizon. Such simple predictions can be remarkably good at producing successful adaptive behavior (Railsback and Harvey 2020).

6.7 Sensing
Both the habitat and activity selection and spawning behavior submodels assume that trout select cells from those within a specific radius, and that they can sense habitat and competition conditions in those cells (Sect. 9.13.1). This habitat selection radius increases with the trout's length. For cells within the radius, trout are assumed able to sense, without uncertainty or error, all cell variables including those representing physical habitat and the availability of resources subject to competition: food, velocity shelter, and hiding places. Trout do not sense anything about cells or habitat outside that radius. The sensing assumptions are based on observations (e.g., Harvey et al. 1999) that salmonids commonly spend at least small amounts of time exploring and frequently alternate among locations in a non-random way; and we assume that the ability to explore increases with swimming ability as trout grow.

6.8 Interaction
InSTREAM has strong interactions among individual trout but they are mostly indirect. Trout compete in a size-based hierarchy for all the resources they consume: food, velocity shelter, and hiding places. This hierarchy is simulated by trout executing their habitat and activity selection submodel in descending order of length; when each trout selects its activity and cell, it consumes resources that are no longer available to the remaining smaller trout.

There is one form of direct interaction: redd superimposition (Sect. 9.36). When one female trout spawns and creates her redd, she can destroy a redd created by a previous spawner.

6.9 Stochasticity
Overall, inSTREAM is not a highly stochastic model: few of its submodels depend mainly on probability parameters and random numbers to determine the timing or outcome of events. However, the model does use stochasticity in several important ways, which can make model results relatively stochastic when modeling small reaches and populations.

The first important use of stochasticity is in the survival submodels (sects. 9.14, 9.31): survival of each trout and egg of each kind of mortality is treated as a random Bernoulli trial (which randomly produces a value of true or false, given the probability of true) with the probability of survival a deterministic function of trout and habitat characteristics. This means that which trout die on each time step is partially random.

Second, whether or not a female trout spawns on a day when all criteria for spawning are met is a random Bernoulli trial. Therefore, the date on which each trout spawns is partially random and this date can strongly affect whether and when the spawner's redd survives to produce new trout. Further, this approach prevents spawning if the last date on which spawning is allowed is reached before the Bernoulli trial is true, so the number of redds produced is also stochastic.

Other less-important uses of stochasticity include setting the initial length and location of trout at the start of a model run (Sect. 7.3), and tie-breaking in selection algorithms: e.g., in the very unlikely event that more than one habitat cell offers the same, best, conditions the trout selects one randomly.

The model parameter ***random-number-seed*** (integer) controls the "random number generator", the software platform's algorithm that produces the pseudorandom numbers used in stochastic processes. If this parameter is given a value of zero, then every model run uses a different sequence of pseudorandom numbers. If instead ***random-number-seed*** is given any non-zero value, that value is used as the generator's "seed": as long as ***random-number-seed*** has the same non-zero value, the same sequence of pseudorandom numbers will be used and—if *all* inputs and parameters remain unchanged—the model will produce exactly the same results. Different values of ***random-number-seed*** produce different pseudorandom number sequences. (The year shuffler uses its own random number seed; Sect. 8.2.)

(NetLogo generates pseudorandom numbers using the "Mersenne Twister" algorithm, a high-quality random number generator used in many modeling platforms.)

6.10 Collectives

InSTREAM does not represent collectives.

6.11 Observation

InSTREAM is a complex model that produces many kinds of results, from primary outputs such as the number and mean size of trout in several age classes once per year, to numerous intermediate outputs such as distributions of age, length, and weight, and the behaviors selected by trout (e.g., how many drift feed, search feed, or hide during day, night, and twilight). Comprehensive output of all potentially useful results can be too extensive to be useful, while too little output (e.g., only the number of fish alive each day) provides too little information to understand the model.

Our solution to the observation concept is to make it easy for users to select the kinds of observation they need. This solution is largely a software issue, addressed in Sect. 14, but here we summarize the approach:

By default, graphical displays show fish locations and habitat conditions (e.g., the depth or velocity of each cell) each time step. However, updating of graphical output can be turned off to reduce execution time.

File output of population summary statistics is normally produced. The user can choose the frequency of this output (e.g., each time step, each simulated day or week) or turn it off.

Summary population output can include a variable *is-census?* which has a value of TRUE only if the current time step is an annual census date. This variable is useful for identifying output comparable to field data collected annually at the same time of year. The census dates are specified by two model parameters: ***census-days*** is a list of day-of-the-year values (e.g., "9/30" for September 30th) that specify the census dates, and ***census-years-to-skip*** specifies the number of simulation years that must be completed before *is-census?* can be TRUE. The value of ***census-years-to-skip*** is typically set to 2-3 years to avoid effects of initial conditions (Sect. 26.3). The value of *is-census?* is TRUE only on time steps that (a) start at a time that is more than ***census-years-to-skip*** years after the simulation start, (b) have the month and day equal to one of the days on the ***census-days*** list, and (c) are the first time step of the day. For example, if a simulation starts on 10/1/2020, ***census-days*** is a list containing "5/1" and "9/30", and ***census-years-to-skip*** is 2, then *is-census?* will be TRUE on the first time step of 9/30/2023 and the first time steps of May 1st and September 30th of subsequent years.

File output of individual-level events is also normally produced: each time a trout experiences an event such as being created, spawning, or dying, it writes a line of output to a file. This file can be used to analyze details such as how many fish, of which ages or sizes, die of each kind of mortality.

Optional “debugging” output can be turned on to produce high levels of detail for specific submodels.

The NetLogo software platform of inSTREAM 7 makes it quite easy to further customize observation, for example by adding graphs updated during execution or by customizing file output.

7 ODD Element 5: Initialization

This ODD element describes, in full detail, how inSTREAM 7 is initialized before the start of a model run[14]. Initialization of inSTREAM 7 is complex, so we describe it in separate stages. Sect. 13 describes the input files used in initialization.

7.1 Observer initialization

The observer’s variables (Table 1) are initialized in the following steps. The value of *sim-time* is set to hour 0 of the date specified by the observer parameter ***start-date***[15]. The values of *prev-time* and *step-length* are not initialized; on the first time step of the simulation, the time updates action sets their value when it advances *sim-time*.

The values of *light-phase* and *sunlight-irradiance* are not initialized but set by the habitat updates action of the first time step.

7.2 Habitat initialization

Initializing inSTREAM 7’s reaches and cells depends heavily on preparation of their characteristics in other software. Geometry (corner coordinates) is typically created in GIS software from field surveys, sometimes with additional information from aerial imagery. Cells are read into inSTREAM 7 as GIS polygons, in the cell setup submodel (Sect. 9.1). The static cell variables (Table 3) are read into inSTREAM from the GIS file as properties of the polygons[16].

[14] Per NetLogo convention, the overall schedule for initialization is in the procedure `setup`.

[15] Initialization of time variables is coded in procedure `set-up-time`.

[16] Creation and initialization of cells is coded in procedure `build-cells`.

The hydraulics submodel (Sect. 9.3) requires two lookup tables for each cell, relating the cell's depth and velocity to reach flow. These tables are typically prepared via parameterization and calibration of a two-dimensional hydrodynamic model, although simpler approaches such as the quasi-two-dimensional PHABSIM hydraulic models can also be used. GIS analysis can be used to re-scale hydraulic model results if the hydraulic model uses a mesh finer than the inSTREAM polygons. The hydraulics setup submodel (Sect. 9.2) builds these lookup tables using data read in from files[17].

After its lookup tables are read in, the hydraulics submodel is executed to set the initial depths and velocities of all cells, for the flow that occurs at the initial value of *sim-time*.

The static variables of reaches (Table 2) are read in from the parameter file described in Sect. 11. The dynamic variables of reaches are not initialized until the habitat updates action of the first time step.

7.3 Trout initialization

InSTREAM simulations start with an initial trout population generated, in the trout setup submodel (Sect. 9.4), from characteristics provided as input (Table 6)[18]. This input includes multiple records (the rows in Table 6), each specifying the trout species, reach that the trout will be created in, age, number of initial trout to create, and minimum, mode, and maximum of the initial length distribution.

Table 6: Example trout population initialization input

Species	Reach	Age	Number	Length min	Length mode	Length max
Rainbow	Eska	0	300	3	4	5
Rainbow	Eska	1	50	8	10	12
Rainbow	Eska	2	10	15	18	22
Brown	Eska	0	500	3	4	5
Brown	Eska	1	100	10	12	14
Brown	Eska	2	50	18	21	25
Rainbow	Belagua	0	60	3	4	5
Rainbow	Belagua	1	10	8	10	12
Rainbow	Belagua	2	2	15	18	22
Brown	Belagua	0	100	3	4	5
Brown	Belagua	1	20	10	12	14
Brown	Belagua	2	10	18	21	25
Rainbow	Belagua	0	12	3	4	5

[17] The code that initializes the hydraulic lookup tables is in procedure `read-hydraulics`.

[18] Creation of the initial trout population is coded in procedure `build-initial-populations`. Trout state variables are initialized in the procedure `initialize-trout-with`.

For each such initialization record, inSTREAM creates the number of trout of the species and age specified by the record, and places them in the specified reach. The age can be any integer; it is not restricted by the age classes used to summarize output (Sect. 2.2.8). Sex (*trout-sex*) is assigned to each individual randomly, with equal probability of male and female. Each initial trout has its length (*trout-length*) drawn randomly from a triangular distribution with the specified minimum, mode, and maximum. Initial trout are assigned a condition (*trout-condition*) of 1.0, meaning that their weight (*trout-weight*) is set to the healthy weight for the initial length (Sect. 9.20).

Each initial trout's location *trout-cell* is set by (a) randomly selecting a location from among all those that are within the reach's wet cells (defined as cells with depth greater than zero after they are initialized), and then (b) setting *trout-cell* to the cell that contains the selected location. In this process, "locations" are combinations of integer X and Y coordinates in NetLogo's built-in grid space: all the points where the GIS coordinates divided by (100 × ***world-resolution***) are both integers. This process makes the probability of a cell being chosen as the initial location of a fish approximately proportional to the cell's area.

The dynamic variable *trout-activity* does not require initialization. The three memory lists of trout are initialized by creating the lists and giving each list one member item. The items on *growth-memory-list* and *consumption-memory-list* are the number 0.0, and the initial item on *survival-memory-list* is the number 1.0. Therefore, on the first time step the habitat and activity selection submodel assumes that the trout have, previously on the same day, consumed zero food and experienced zero growth (even though those two assumptions conflict because zero consumption causes negative growth; Sect. 9.20), and experienced a survival probability of 1.0.

The variable *trout-spawned-this-season?* is initialized to FALSE. This value allows initial trout to spawn in the first simulated year if simulations start during the spawning season (Sect. 9.27.2).

The value of *trout-superind-rep* is initialized to 1: no initial trout are superindividuals.

7.4 Observer output

The last initialization event is to execute the action that updates model outputs (the last action on the schedule; Sect. 5). This output reflects the state of the model before simulations begin.

8 ODD Element 6: Input Data

Time-series input drives many of the dynamic habitat variables: on each time step, the habitat update submodel (Sect. 9.8) reads in values of flow, temperature, and turbidity for each reach. The flow value then drives the hydraulics submodel's update of cell depths and velocities (Sect. 9.3).

8.1 Flow, temperature, and turbidity input

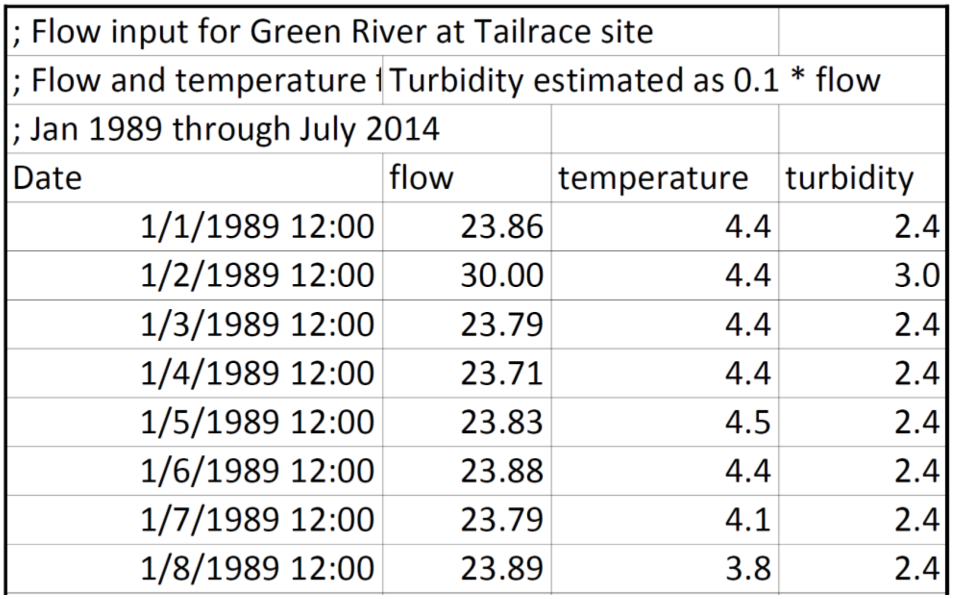
The input for flow, temperature, and turbidity of each reach are read from an input file that includes, in each row, a date and time and corresponding values of flow, temperature, and turbidity (Sect. 13.5)[19]. The interval between input values is not restricted: it can be (for example) one day, one hour, one week, or variable (to capture short-term flow events). (However, see Sect. 14.1.7 concerning how the inSTREAM software checks for missing input

[19] Reading of new time series variable values is coded in procedure `update-habitat`.

values.) The input value used for each time step is obtained by finding the input record with date-time value closest to the date and time of the *middle* of the time step. Because of this method for assigning time series input to each time step, input in the form of daily mean values should be entered in the time series input file with a time of 12:00 (as in Figure 3) so all of a day's time steps use the same mean value.

The middle of a night time step is always be exactly midnight (Sect. 9.6). When daily input has a time of 12:00, inSTREAM uses the input for the day starting (not ending) at midnight. In other words, when daily input with a time of 12:00 is used, night time steps are assigned the flow, temperature, and turbidity of the calendar day on which the time step ends (e.g., the time step extending from 19:30 on May 10 to 4:30 on May 11 uses input for 12:00 of May 11).

Even though inSTREAM 7 does not require time series input at any specific or consistent interval, its software enforces two restrictions. First, the input file must include values from times within 4.0 days of (a) midnight at the start of the first simulated day (parameter ***start-date***) and (b) 23:00 of the last simulated day (***end-date***). Second, the software checks for missing dates in the input file (which are an extremely common mistake), using methods described at Sect. 14.1.7.



; Flow input for Green River at Tailrace site			
; Flow and temperature f	Turbidity estimated as 0.1 * flow		
; Jan 1989 through July 2014			
Date	flow	temperature	turbidity
1/1/1989 12:00	23.86	4.4	2.4
1/2/1989 12:00	30.00	4.4	3.0
1/3/1989 12:00	23.79	4.4	2.4
1/4/1989 12:00	23.71	4.4	2.4
1/5/1989 12:00	23.83	4.5	2.4
1/6/1989 12:00	23.88	4.4	2.4
1/7/1989 12:00	23.79	4.1	2.4
1/8/1989 12:00	23.89	3.8	2.4

Figure 3: Example time series input, viewed in a spreadsheet. (Sect. 13.5 describes the file format.)

Turbidity data are often lacking for inSTREAM study sites, and for previous versions it was common to simply input values of zero turbidity for clear streams. However, the cell light submodel of inSTREAM 7 (Sect. 9.10) calculates light attenuation with depth as a function of turbidity and assumes no attenuation when turbidity is zero. **To avoid unrealistic irradiance values, avoid use of zero values for turbidity**; a good alternative in the absence of data is to assume a proportional relation between flow and turbidity (turbidity = Z × flow, where Z is a constant).

8.2 The year shuffler

InSTREAM 7, like previous versions, includes a “year shuffler” tool that optionally randomizes the sequence of input years[20]. The year shuffler provides one way to replicate simulations. The standard replication method, using different sequences of pseudorandom numbers (Sect. 6.9), illustrates the variability in the model’s output that results from its stochastic processes. Replication via the year shuffler instead illustrates the variability due to year-to-year variation in hydrologic and weather conditions.

The input years to be shuffled are not calendar years but instead “water years”: years that start on a date when stream flows are typically at their lowest and steadiest. In the United States, the convention used by the US Geological Survey starts water years on 1 October. However, the year shuffler lets users define the water year’s start date. Also by convention, water years are numbered by the calendar year in which they end: if the water year starts on 1 October, then any date between 1 October 2007 and 30 September 2008 belongs to water year 2008.

The year shuffler uses three model parameters. The first, ***shuffle-years?***, is a true-false variable that turns the shuffler on or off. If ***shuffle-years?*** is set to false, then the model uses the time series input exactly as provided in the input file. The parameter ***shuffle-day*** is the day of the year (input as a character string in M/d format, e.g., `"10/1"`) on which new water years start.

The third shuffler parameter is its random number seed, ***shuffle-rand-seed***, an integer. If the value of ***shuffle-rand-seed*** is zero the year shuffler uses the model’s main random number generator (Sect. 6.9). Therefore, input will be shuffled in a different sequence every model run if the model parameter ***random-number-seed*** is set to zero. However, if ***shuffle-rand-seed*** is zero and ***random-number-seed*** has a non-zero value, the sequence of shuffled years may or may not be unchanged among model runs: if any change in parameter values or inputs (e.g., the number of initial fish) causes the random number generator to be used a different number of times before the years are shuffled, then the year sequence will be different. Therefore, if ***shuffle-rand-seed*** has a non-zero value the year shuffler uses its own random number generator that uses ***shuffle-rand-seed*** as its seed. Whenever ***shuffle-rand-seed*** has a non-zero value, the year shuffler will produce exactly the same sequence of shuffled water years in the input data, and changing the value of ***shuffle-rand-seed*** changes that shuffled sequence.

When the year shuffler is turned on, it re-writes the time series input for each reach using the following steps.

A list of calendar years in the simulation is created. This “year list” contains the calendar year (e.g., 2007) for each year from the simulation’s start to end (from ***start-date*** to ***end-date***). For example, a model run starting 9/15/2015 and ending 11/30/2020 would produce a list containing 2015, 2016, 2017, 2018, 2019, 2020. (The year shuffler obviously cannot be used for model runs that start and end in the same calendar year.)

A “shuffled year list” is created by copying the year list and randomly shuffling it.

The time series input is re-written. For each row, a year-shuffled time value is created by copying the real time and replacing its year with a shuffled one. If the day of the year is before ***shuffle-day***, then the shuffled year is the corresponding year in the shuffled year list: if the year of the

[20] The year shuffler is programmed entirely within the procedure `set-up-time`. When the year shuffler is turned on, it writes an output file containing the shuffled input and the time from which shuffled values were taken.

input's date is (e.g.) the third year on the year list, then the shuffled year is the third year on the shuffled year list. If instead the date is on or after ***shuffle-day***, the shuffled year is the next year in the shuffled year list (e.g., if the true year is third on the year list, the shuffled year is the fourth item on the shuffled year list). Then the flow, temperature, and turbidity values for the input time are replaced by those corresponding to the year-shuffled time value. This algorithm causes the input to be taken from a random new year each time ***shuffle-day*** is reached.

The year shuffler must cope with leap years: replacing the input for 2004 with input from 2007 must deal with the fact that 2004 includes a 29 February and 2007 does not. When the year shuffler copies input from a non-leap year into a leap year, the values for 29 February are those of 28 February in the non-leap year. When input from a leap year are copied into a non-leap year, the 29 February values are ignored.

9 ODD Element 7: Submodels

Element 7 of ODD provides complete detail for the submodels identified in the "overview" elements. The submodels are presented in approximately the order in which they are executed during (a) initialization and (b) each time step, with related submodels grouped together.

9.1 Cell setup

The cell setup submodel creates the cells of each reach and sets their state variables, as part of initialization (Sect. 7.2)[21]. Cells are input as polygons. The centroid location of each cell is calculated from the polygon vertex (corner) coordinates, and the cell's location set to its centroid location. The static cell variables (*cell-area*, *cell-num-hiding-places, cell-escape-dist, cell-frac-vel-shelter, cell-frac-spawn*) are also read from an input file. Then the hydraulics setup submodel (Sect. 9.2) is executed. Dynamic cell state variables are initialized during the habitat updates action of the first time step.

(All these cell characteristics are read from an ESRI shapefile, as explained in Sect. 13.2. The software performs many additional steps during cell setup, such as drawing the polygons on the display, identifying each cell's adjacent cells, and creating data structures that increase execution speed.)

9.2 Hydraulics setup

This submodel reads in the lookup tables used by the hydraulics submodel (Sect. 9.3) to calculate cell depths and velocities from flow[22]. These tables are input in the form of two files, one each for depth and velocity. A separate pair of files is input for each habitat reach. Each file includes (a) a list of flows at which depth/velocity was calculated, and (b) a table with the depth/velocity of each cell at each of the flows (Figure 4). These input files must be prepared in hydraulic modeling software, independently of inSTREAM.

The hydraulics setup submodel creates separate depth and velocity lookup tables for each cell. To create the depth lookup tables, it first reads in the list of flows at which depth is provided, and then the list of corresponding depths for each cell. The depth lookup table for each cell is created from this input: the table is created with a "point" (pair of flow and corresponding depth values) for each flow and cell depth in the input file. Each cell's depth table is then modified to remove

[21] Cell setup code is in the procedure `build-cells`.

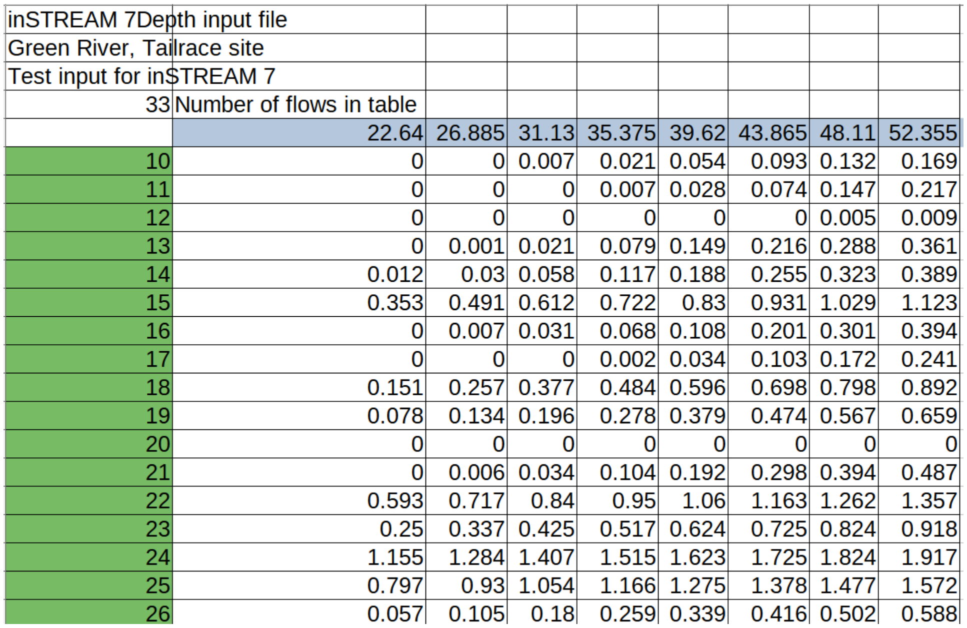
[22] Hydraulics setup is coded in procedure `read-hydraulics`.

flows at which depth is zero, but to include either (a) the flow above which the cell has positive depth, which is also the value of the cell variable *cell-flow-at-wetting*; or (b) the depth when flow is zero, for cells that are always wet even when there is no flow (e.g., in pools). These steps are used:

- If the first flow in the input file is 0.0, then none of the following steps are used. The following steps determine how depth is calculated at flows between zero and the lowest in the input.
- If depth is positive at the lowest flow in the lookup table input, and depth decreases between the first and second flow with non-zero depth (unusual but possible) or is equal at those two flows, then depth at zero flow is assumed equal to depth at the first input flow. A point is added to the lookup table with flow = 0.0 and depth equal to the first input depth; *cell-flow-at-wetting* is set to 0.0.
- Otherwise, the flow at which depth reaches zero ($Q_{0\text{-}E}$) is extrapolated backwards, linearly, from the two lowest flows that have non-zero depths.
    - If $Q_{0\text{-}E}$ is less than the highest flow in the input having depth of zero, then the lookup table is not modified and *cell-flow-at-wetting* is set to the highest input flow with depth of zero. (This case is not possible if depth is positive at the first input flow.)
    - Otherwise, if depth is positive at the first input flow and $Q_{0\text{-}E}$ is negative, then the depth at zero flow is calculated as the Y-intercept of a line through the two lowest points in the lookup table; *cell-flow-at-wetting* is set to 0.0.
    - Otherwise ($Q_{0\text{-}E}$ is positive and higher than any other input flows), the first point in the lookup table is set to flow = $Q_{0\text{-}E}$ and depth = 0.0; *cell-flow-at-wetting* is set to $Q_{0\text{-}E}$.
    - Any points with flow less than *cell-flow-at-wetting* are removed from the table.

If a cell's depth lookup input includes only one positive depth, at its highest flow, then *cell-flow-at-wetting* is set to the second-highest flow in the input.

This process is repeated for the velocity input, with one difference: the first point in each velocity lookup table always has flow = *cell-flow-at-wetting* and velocity = 0.0 (cells with positive depth at zero flow have velocity approaching zero as flow approaches zero).



inSTREAM 7Depth input file								
Green River, Tailrace site								
Test input for inSTREAM 7								
33	Number of flows in table							
	22.64	26.885	31.13	35.375	39.62	43.865	48.11	52.355
10	0	0	0.007	0.021	0.054	0.093	0.132	0.169
11	0	0	0	0.007	0.028	0.074	0.147	0.217
12	0	0	0	0	0	0	0.005	0.009
13	0	0.001	0.021	0.079	0.149	0.216	0.288	0.361
14	0.012	0.03	0.058	0.117	0.188	0.255	0.323	0.389
15	0.353	0.491	0.612	0.722	0.83	0.931	1.029	1.123
16	0	0.007	0.031	0.068	0.108	0.201	0.301	0.394
17	0	0	0	0.002	0.034	0.103	0.172	0.241
18	0.151	0.257	0.377	0.484	0.596	0.698	0.798	0.892
19	0.078	0.134	0.196	0.278	0.379	0.474	0.567	0.659
20	0	0	0	0	0	0	0	0
21	0	0.006	0.034	0.104	0.192	0.298	0.394	0.487
22	0.593	0.717	0.84	0.95	1.06	1.163	1.262	1.357
23	0.25	0.337	0.425	0.517	0.624	0.725	0.824	0.918
24	1.155	1.284	1.407	1.515	1.623	1.725	1.824	1.917
25	0.797	0.93	1.054	1.166	1.275	1.378	1.477	1.572
26	0.057	0.105	0.18	0.259	0.339	0.416	0.502	0.588

Figure 4: Example extract from a hydraulic input file, viewed in a spreadsheet. The row highlighted in blue provides the flows at which depths were calculated by an external hydraulic model, and the first column (green cells) indicate which cell the depth values are for. For example, cell 46 has a depth of 0.33 cm when the flow is 26.9 m^3/s.

9.3 Hydraulics

The hydraulics submodel is applied to each cell each time flow changes, to update *cell-depth* and *cell-velocity*.[23] It uses the lookup tables created by the hydraulics setup submodel (Sect. 9.2). Here, “flow” refers to the flow for which depth and velocity are calculated, and “depth” and “velocity” refer to *cell-depth* and *cell-velocity*.

This submodel uses linear interpolation; when we say that (e.g.) the velocity for flow *Q* is interpolated from two flows in the lookup table, we mean that the model first fits a line to two points. The first point has X = the first flow (which is less than *Q*) and Y = the velocity in the table for that flow, and the second point has X = the second flow (which is greater than *Q*) and Y = the velocity for the second flow. Then the velocity for *Q* is calculated as the Y value on this line corresponding to X = *Q*.

The submodel uses these steps:

When flow is above *cell-flow-at-wetting* (the lowest lookup table flow, above which depth is positive) and below the highest flow in the table, depth and velocity are interpolated from the lookup table points just below and above the flow.

If instead flow is equal to or below *cell-flow-at-wetting*, depth and velocity are set to zero.

If the flow is higher than the highest flow in the lookup tables then both depth and velocity are extrapolated linearly upward from their values at the two highest flows in the table. (The extrapolation method is identical to interpolation except that the flow is higher than, not between, the two lookup table points.) This extrapolation for higher flows does not allow cells dry at the

[23] The hydraulics submodel is coded in the procedure `update-hydraulics-for`.

highest flow in the lookup table to become wet at even higher flows. It is possible for this extrapolation to produce negative depths or velocities, when a cell has a lower depth or velocity at the highest flow in the table than at the second-highest flow. In this surprisingly common case, negative depths are set to zero but negative velocities raise an error; the problem must be solved by revising the lookup table.

To prevent these interpolation methods from having strong and unrealistic effects, it is very important for the depth and velocity lookup tables to include many flows spanning the range to be simulated in inSTREAM. Doing so often requires extrapolating a hydraulic model's calibration beyond its recommended range, but extrapolation in the hydraulic model is preferable to extrapolation in inSTREAM.

9.4 Trout setup

This submodel creates the initial trout population at the beginning of a model run[24]; Sect. 7.3 describes it fully. (The method for drawing a random length from a triangular distribution was taken from: en.wikipedia.org/wiki/Triangular_distribution#Generating _Triangular-distributed_random_variates, accessed 27 November 2015.)

9.5 Day length

This submodel calculates the length (d) of daytime *day-length* and of twilight (the two times per day when the sun is between the horizon and 6° below the horizon; *twilight-length*). These values are functions of the date and site latitude.[25]

InSTREAM uses the day length method of Glarner (2011), in particular Glarner's equations 7, 8, 14, and 15 for day length and equations 23 and 27 for day length including the two twilight periods. The observer variable *day-length* is set to day length without twilight, and *twilight-length* (the length of one twilight) is set to one half of the difference between day length with twilight and *day-length*. The twilight angle (model parameter ***twilight-angle***) has a standard value of 6°, in keeping with a standard definition of twilight called "civil twilight."

The inSTREAM implementation assumes that the winter solstice is always on December 21, so the day used in Glarner's equations (day of the solar year, starting on the winter solstice) is calculated as the Julian date plus 10 d. Because inSTREAM assumes there is always a morning and evening twilight period, it cannot be used at polar latitudes (> ~66°) where the entire day can be either daytime or night. The model software raises an error condition if *day-length* is calculated to be ≤0.0 or ≥1.0 or if *twilight-length* is ≤0.0.

Figure 5 illustrates the behavior of this submodel, including that the length of dawn and dusk phases varies somewhat with latitude but little (compared to variation in day and night phases) with date.

[24] Trout initialization is coded in the procedure `build-initial-populations`, which uses the procedure `initialize-trout-with` to actually initialize each new fish.

[25] The day length calculations are coded in the procedure `update-day-length`.

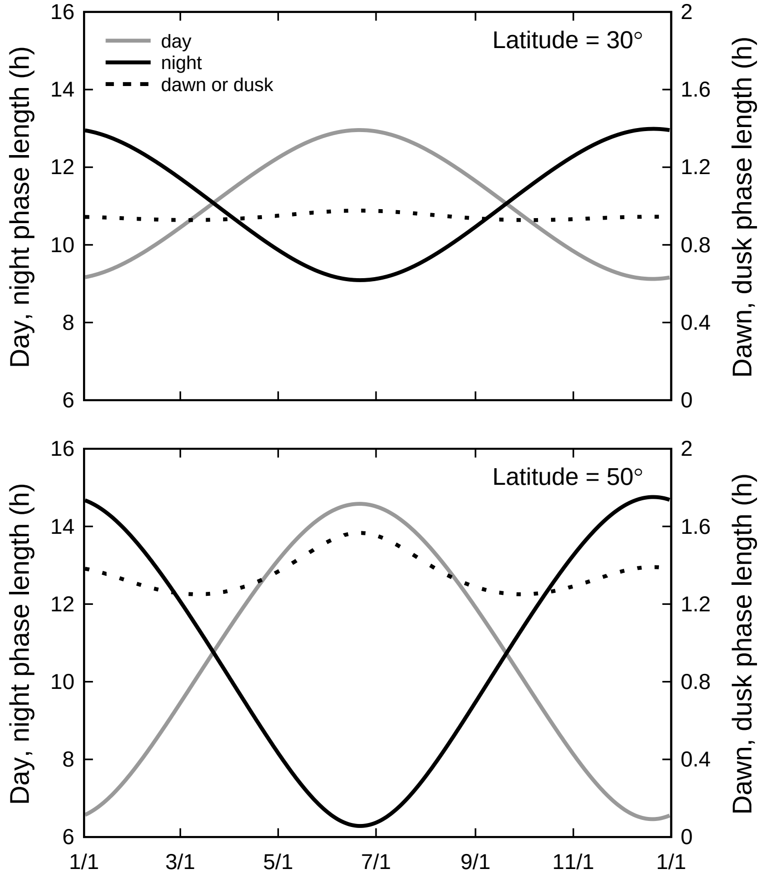


Figure 5: Duration (h) of day, night, and twilight (both dawn and dusk) phases over a year, for (top) latitude = 30°, and (bottom) 50º. Twilight phase length uses the right Y axis.

9.6 Step timing

The step timing submodel calculates the start and end times of a day's time steps[26]. It executes at the start of the first time step of each calendar day (whenever the previous time step ended on a different date than it started). Time step lengths are determined by sunlight: new time steps start at the beginning of the dawn, day, dusk, and night phases. (InSTREAM 7 could be modified to use additional variables, such as flow or temperature, also trigger new time steps.)

The following steps are used.

The day length submodel (Sect. 9.5) is used to update the value of *day-length* and *twilight-length* for the new date.

The end of the current time step (a night phase) is the start of the dawn phase, assumed to be when the sun is 6° below the horizon. The time (d) between midnight and the start of dawn is calculated as [(1.0 – *day-length*) / 2] – *twilight-length*.

[26] This submodel is programmed in the procedure `update-time-and-habitat`.

The end of the following dawn phase (and start of day) is assumed to be when the sun is 6° above the horizon, which is the time of sunrise plus *twilight-length*. The time between midnight and the start of day is calculated as the start of dawn plus (2 × *twilight-length*).

The end of the day phase (start of dusk) is assumed to be when the sun falls to 6° above the horizon, the time of which is equal to the start of dawn plus *day-length*.

The end of the dusk phase (start of night) is when the sun falls to 6° below the horizon, calculated as the time at which dawn ends plus *day-length*.

After these calculations, a series of checks are made to ensure that each light phase has a non-zero duration. At high latitudes (~62-66°N), the day length and step timing methods do not work on dates near the solstices; for example, the duration of night plus dawn plus dusk can exceed 1.0 days. The following checks allow inSTREAM to be used at these high latitudes, though they produce somewhat artificial phase lengths (and therefore light intensities; Sect. 9.9) near the solstices. The checks ensure that each of the four daily light phases is at least 0.002 d (3 minutes) long, every day: If dawn starts less than 0.001 d after midnight, its start (and the end of the previous night) is changed to 0.001 d after midnight. If dawn ends later than 0.001 d before noon, its end (and the start of day) is changed to 0.001 d before noon. If dusk starts earlier than 0.001 d after noon, its start (and the end of day) is set to 0.001 d after noon. If dusk ends later than 0.001 d before midnight, its end (and the start of night) is set to 0.001 d before midnight.

9.7 Simulation time

This submodel updates the model's time variables, and includes the "stopping rule" that ends a simulation[27]. It uses the results of the time stepping submodel. The value of *sim-time* is advanced to the end of the next phase (if *light-phase* was night, *sim-time* is set to the time at which dawn ends; if *light-phase* was dawn, *sim-time* is set to the time at which day ends; etc.) and the value of *prev-time* is set to the previous value of *sim-time*. The value of *light-phase* is advanced to the next phase (from night to dawn, dawn to day, etc.). The value of *step-length* is set to the difference between *sim-time* and *prev-time*.

The stopping rule checks to see whether the new value of sim-time is after the 23rd hour of the date specified by model parameter ***end-date***[28]. If so, the model stops immediately. Therefore, simulations include phases through dusk of ***end-date***.

9.8 Habitat update

The habitat update submodel updates the reach and cell habitat variables each time step[29], mainly by calling other submodels. It executes these steps:

Time series inputs are updated as described in Sect. 8.

The hydraulics submodel (Sect. 9.3) is called to update cell depths and velocities.

The surface light submodel (Sect. 9.9) calculates the surface irradiance for the time step.

The cell light submodel (Sect. 9.10) updates the irradiance in each cell.

[27] The simulation time submodel is also coded in the procedure `update-time-and-habitat`.

[28] The stopping rule is programmed in the `go` procedure.

[29] The habitat update submodel is programmed in the procedure `update-habitat`.

The availability of cell resources such as food and cover are updated by the cell resource depletion submodel (Sect. 9.11).

9.9 Surface light

The surface light submodel calculates the mean sunlight irradiance (observe variable *sunlight-irradiance*, abbreviated here as *SI*) for each time step[30]. In general, inSTREAM 7 users need not be overly concerned about precise simulation of light because its effects on trout feeding (Sect. 9.22) and survival (sects. 9.18, 9.19) are not highly sensitive to small variation in irradiance.

The submodel is designed to be general and applicable for time steps of any length and starting and ending at any time of day. It includes two components: a function that calculates mean irradiance between two times of day (while the sun is continuously above the horizon); and a set of equations that calculate time-weighted average irradiance for a time step, depending on how much of the time step occurs during night vs. day. The mean irradiance function is referred to hear as LIB(*time-A*, *time-B*) where *time-A* and *time-B* are the start and end times of the period for which light is calculated.

Unlike the step timing submodel (Sect. 9.6), this surface light submodel does not consider twilight explicitly but instead treats sunrise and sunset as the start and end of daytime. This difference is because the available irradiance models address direct radiation, which is zero during twilight. The methods we use to approximate light during dawn and dusk are explained below. **In this submodel description only, we ignore twilight and delineate "day" and "night" by sunset and sunrise.**

In this submodel description, observer date and time variables such as *step-length* and *day-length* refer to the time step and date for which irradiance is being calculated. The submodel also uses *sunrise-time* and *sunset-time*, the date and time of the current day's sunrise and sunset. *Sunrise-time* is set to noon minus one half of *day-length*, and *sunset-time* is set to *sunrise-time* plus *day-length*. When the sunrise and sunset times and day length of the following day are required, they are approximated as the same as for the current day.

This submodel assumes that irradiance at night is constant at the value of the observer parameter ***light-at-night***, which can be adjusted to reflect typical nighttime light levels of a study site and their effect on fish feeding (Sect. 9.22). Our default value is 0.9 W/m^2.

9.9.1 Function for mean irradiance over a period of daylight

This function LIB(*time-A*, *time-B*) assumes that the time period between *time-A* and *time-B* is of continuous daylight on the same day. The function is modified from equation 1.10.4 of Duffie and Beckman (2013). Mean solar irradiance above the water surface (*LIB*, W/m^2) between two times (*t1* and *t2*) is approximated as:

$$LIB = C\frac{12G_{SC}}{\pi(t2-t1)}\left(1 + 0.033cos\left(\frac{360n}{365}\right)\right)\left(cos\varphi cos\delta(sin\omega 2 - sin\omega 1) + \frac{\pi(\omega 2-\omega 1)}{180}(sin\varphi sin\delta)\right).$$

[30] Surface light updates are coded in procedure `update-light`. Calculation of mean irradiance between two times during the day is coded in `light-intensity-between`.

This equation approximates mean irradiance (W/m^2) on an unshaded horizontal surface that is not under water, between times *t1* and *t2* (corresponding to *time-A* and *time-B*). The variable *C* is an observer parameter (***light-correction***) with a value between zero and one, reflecting the extent to which extra-terrestrial radiation is reduced by the atmosphere, clouds and humidity, and pollution. Duffie and Beckman (2013) provide methods and data (e.g., their Table 2.7.2) for estimating *C*. For relatively cloud-free areas, values in the range of 0.6 to 0.8 appear reasonable for *C*. The variable G_{SC} is the solar constant, assumed to be 1367 W/m^2 (Sect. 1.2 of Duffie and Beckman 2013). The variable *n* is the Julian date (day of the year, starting with 1 on January 1). The variables $\omega 1$ and $\omega 2$ are the "hour angles" (degrees) for times *t1* and *t2*, equal to -180 + (15°×*t*) where the time *t* is in hours.

The variable φ is the latitude (° north), the observer parameter ***latitude***. The variable δ is the solar declination (angle of the sun above the equator at noon), updated in the day length submodel as:

$$\delta = 23.45 \times sin\left(360\frac{284 + n}{365}\right).$$

9.9.2 Equations for time step mean irradiance

The following algorithms are used to calculate *SI* depending on when the time step begins and ends.

<u>If the entire time step is at night</u> then *SI* = ***light-at-night***.

<u>If the time step spans from night to day</u> (which includes dawn phases), irradiance is approximated as the time-weighted average of night and day values:

$$SI = \frac{lightAtNight \times (sunriseTime - prevTime) + LIB(sunriseTime, simTime) \times (simTime - sunriseTin}{stepLength}$$

For a dawn phase time step, which has equal amounts of time before and after sunrise, this equation reduces to:

$$SI = \frac{lightAtNight + LIB(sunriseTime, simTime)}{2}.$$

If the time step is entirely within daytime (as in a daytime phase), then:

$$SI = LIB(prevTime, simTime).$$

If the time step goes from daytime to night (as in a dusk phase):

$$SI = \frac{LIB(prevTime, sunsetTime) \times (sunsetTime - prevTime) + lightAtNight \times (simTime - sunsetTime)}{stepLength}.$$

The results of this submodel are illustrated by Figure 6, which shows mean surface irradiance over the year for two latitudes, during daytime and twilight phases. Note that during dawn and dusk phases irradiance is always within a narrow range of ~21-26 W/m^2, while daytime light varies within a year by ~200-300 W/m^2.

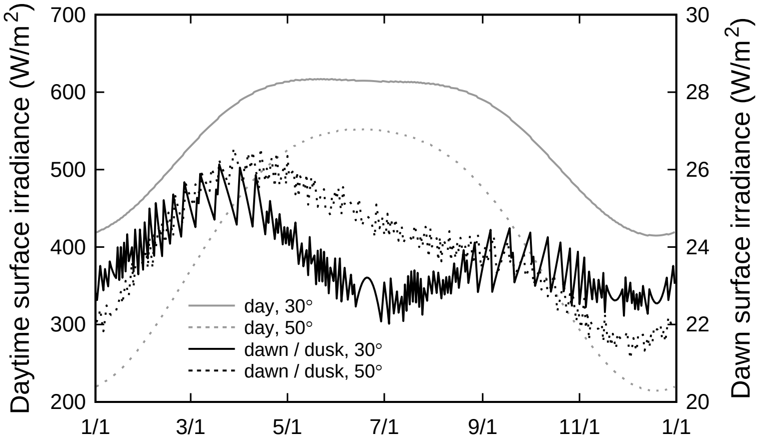


Figure 6: Mean surface irradiance as a function of date, during daytime (left Y axis) and twilight (dawn and dusk, which have equal irradiance; right Y axis), for latitudes of 30° and 50°. The oscillations in twilight values are due to surface irradiance being a complex function of phase length and sun angles through the day; these oscillations are negligible (<2 W/m^2; right Y axis).

9.10 Cell light

This submodel calculates, for each cell, a light irradiance value that represents what fish occupying the cell experience. Considering that fish are not always on the cell bottom, and that they are affected more by light conditions above than below them (for seeing food, or being seen by predators), we define the characteristic irradiance as that occurring at half the cell's depth[31].

The submodel assumes that cell irradiance varies with the surface irradiance (Sect. 9.9), cell depth, turbidity, and local shading (Julian et al. 2008). (Julian et al. 2008 also include a coefficient reducing irradiance by the reflection at water surface; we neglect that term as minor and indistinguishable from shading.) We use the standard Beer's Law approach, which models irradiance as a negative exponential function of depth, with the rate of light attenuation with depth depending on the optical qualities of the water:

$$cellLight = sunlightIntensity \times reachShading \times exp(-k \times cellDepth/2).$$

In this equation, ***reach-shading*** is a reach parameter that represents the fraction of solar irradiance that is *not* blocked by (e.g.) vegetation and topography. Julian et al. (2008) discuss methods for estimating ***reach-shading***; shading has been modeled as a function of local horizon angles and date, e.g., in the water temperature models of Theurer et al. (1984). This parameter is only a coarse overall adjustment factor, because inSTREAM represents shading at the reach scale and ignores variation over time (both within-days and across seasons).

The light extinction coefficient *k* increases with turbidity. A number of empirical models relating *k* to turbidity have been published; variation among them is likely due to variation in the cause of turbidity (algae vs. mineral; type of mineral) and how it refracts and absorbs light. We use this simple equation for *k*:

$$k = (lightTurbidCoef \times turbidity) + lightTurbidConst$$

[31] The cell light submodel is programmed in the procedure `update-habitat`. The assumption that the characteristic light level is that at half the cell's depth is programmed there but easily changed.

where *turbidity* is a reach state variable (Sect. 4.1.2) and ***reach-light-turbid-coef*** (cm^{-1}/NTU) and ***reach-light-turbid-const*** (cm^{-1}) are reach parameters. We use the values of Julian et al. (2008) as standard values; these values and alternatives are in Table 7. (Note that the literature usually reports *k* in units of m^{-1} but we convert the models to units of cm^{-1}.)

Table 7: Alternative parameter values for light extinction

Source	Basis	***reach-light-turbidity-coef***	***reach-light-turbid-const***
Figure 5 of: Julian et al. (2008)	Data from a small spring-fed creek and a large warmwater river	0.0017	0.0
Equations 3 and 4 of: Hansen and Beauchamp (2015), fit to a linear model	Data of Lloyd et al. (1987) from 14 cold Alaskan lakes	0.0045	0.039
Equation 2 of: Lloyd et al. (1987)	Data from Eklutna Lake, Alaska	0.00064	-0.00093
Equation 4 of: Lloyd et al. (1987)	Data from Alaskan streams with turbidity from placer mining	0.00024	0.01

This submodel causes irradiance to decrease substantially with cell depth, even at relatively low turbidity values (Figure 7). However, a turbidity of zero produces no light attenuation with depth. Users can prevent an unrealistic lack of light attenuation by avoiding zero values for turbidity.

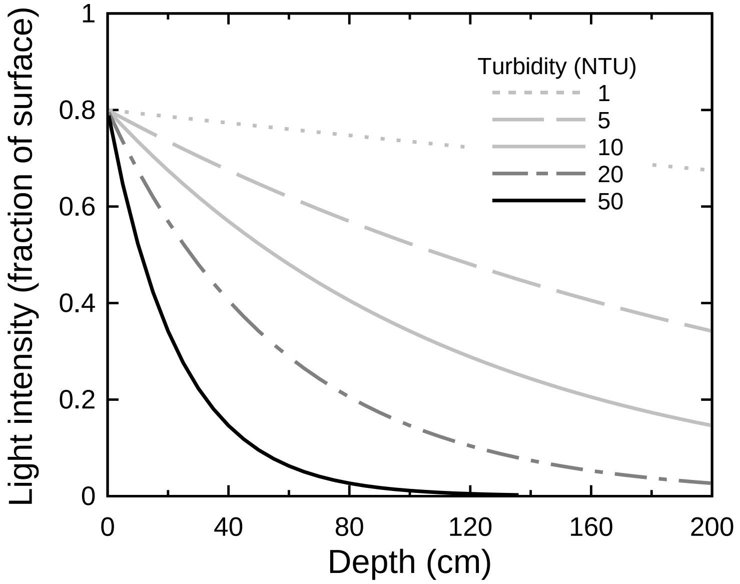


Figure 7: The cell light submodel's light attenuation as a function of depth and turbidity. The Y axis is the degree to which surface light is attenuated at half the cell's depth, or cell-light / sunlight-irradiance. The shading parameter is equal to 0.8 (light is reduced 20% by shading).

9.11 Cell resource depletion

This submodel does "bookkeeping" to model the availability of cell resources depleted by fish within each time step. As the habitat and activity selection action (Sect. 9.13) is executed, each trout considers how much of each such resources is available in each cell when deciding which cell to occupy, and then depletes some of the resources of the cell it occupies. Subsequent fish cannot use the depleted resources. The submodel has two separate functions: resetting resources to undepleted levels as part of the habitat update action at the start of each time step, and tracking resource depletion by trout during the habitat and activity selection action[32].

9.11.1 Resetting resources at the start of a time step

At the start of each time step, the submodel resets depletable cell resource variables to undepleted levels. The two food resource variables *cell-available-drift* and *cell-available-search* represent rates at which food becomes available to the trout (g/d).

The rate at which drift food is available is modeled as the volume of water flowing into the cell (cm^3/d) times a concentration of drift food (g/cm^3), plus a term representing regeneration of food consumed by trout within the cell. The undepleted value is modeled as:

$$cellAvailableDrift = \frac{86400 \times cellArea \times cellDepth \times cellVelocity \times reachDriftConc}{reachDriftRegenDist}$$

where 86400 s/d converts the rate from per second to per day, and *cell-area* (cm^2), *cell-depth* (cm), and *cell-velocity* (cm/s) are cell state variables. ***Reach-drift-conc*** (g/cm^3) and ***reach-drift-regen-distance*** (cm) are reach parameters. These two parameters strongly affect availability of the most important trout food resource and, therefore, trout growth, size, and abundance. The value of ***reach-drift-conc*** is typically estimated via calibration (Sect. 24.3), which commonly produces values in the range of $1\text{-}10\times10^{-10}$. ***Reach-drift-regen-distance*** can be thought of as the distance over which drift consumed by fish is replenished from the benthos; lower values represent more drift-productive conditions. This parameter can also be estimated via calibration because lower values allow more simulated trout to occupy the same cell without changing the food intake rate obtained by each trout. Values for ***reach-drift-regen-distance*** commonly range 500-2000 cm.

The rate at which search food becomes available is modeled very simply, given the potential complexity of search feeding and lack of literature on it. The undepleted value is reset to:

$$cellAvailableSearch = cellArea \times reachSearchProd$$

where ***reach-search-prod*** (g/cm^2/d) is a reach parameter representing the rate at which search food becomes available. This parameter can also be estimated via calibration; it generally affects growth of only small trout (Sect. 24.3). Values in the range of $1\text{-}4\times10^{-5}$ are typical. (Previous

[32] The resource resetting function is programmed in the procedure `update-habitat`. Depletion of resources is programmed in `deplete-drift-by`, `deplete-search-by`, `deplete-vel-shelter-by`, and `deplete-hiding-places-by`.

versions of inSTREAM treated this parameter as an hourly rate, so values from previous versions must be multiplied by 24 before use in inSTREAM 7.) This parameter cannot be directly estimated from measurements of benthic biomass because (a) there is no clear relation between biomass density and the rate at which it becomes available as food and (b) it also represents other food sources with availability independent of flow, such as terrestrial invertebrate drop into pools.

Two other depleted cell resources represent the cell area providing two kinds of "cover". The variable *cell-available-vel-shelter* represents the available area that provides velocity shelter for drift-feeding fish. (Sect. 22.2 provides guidance on estimating *cell- available-vel-shelter*.) It has an undepleted value of:

$cellAvailableVelShelter = cellArea \times cellFracVelShelter.$

The variable *cell-available-hiding-places* represents the number of hiding places remaining that adult trout can occupy while using the hiding activity instead of feeding (Sect. 9.13). It is reset to an undepleted value of the cell variable *cell-num-hiding-places*.

9.11.2 Depletion of resources during habitat and activity selection

After each model trout selects the habitat cell and activity it uses on the current time step, the resources it uses are subtracted from those remaining for other fish that consider using the selected cell, using the following steps. The variables discussed here are for the trout that selected its habitat and activity and for the cell that trout chose to occupy.

If the selected activity is "hide" then no food resources are depleted; *cell-available-drift* and *cell-available-search* remain unchanged. Likewise, *cell-available-vel-shelter* is unchanged. If the number of unoccupied hiding places *cell-available-hiding-places* is greater than zero, it is decreased by one and the individual is assumed to use a hiding place. (For superindividuals—trout with value of *trout-superind-rep* > 1—the number of hiding places is decreased by the trout's value of *trout-superind-rep*.) If *cell-available-hiding-places* is zero (or less than a superindividual's value of *trout-superind-rep*) then the individual is assumed unable to use a hiding place (Sect. 9.18.6) and *cell-available-hiding-places* is unchanged.

If the selected activity is "drift" then *cell-available-hiding-places* and *cell-available-search* are unchanged, but the availability of drift food and velocity shelter are decreased. The value of *cell-available-drift* is updated by subtracting the drift food intake of the trout, which is calculated in the habitat and activity selection submodel. (That submodel limits drift food intake so it cannot exceed *cell-available-drift*, so the value of *cell-available-drift* cannot become negative.) The habitat and activity selection submodel assumes that trout selecting the "drift" activity occupy velocity shelter if enough is available in the cell; "enough available" is defined as *cell-available-vel-shelter* ≥ *trout-length*^{2}. Therefore, the value of *cell-available-vel-shelter* is decreased by the square of the trout's length; if the result would be negative then the trout does not use velocity shelter and *cell-available-vel-shelter* is unchanged (so velocity shelter remains potentially available for smaller trout). (The amount of drift food and velocity shelter depleted by a superindividual trout is multiplied by its value of *trout-superind-rep*.)

If the selected activity is "search" then *cell-available-hiding-places*, *cell-available-drift*, and *cell-available-vel-shelter* are unchanged and the availability of search food is decreased. The value of *cell-available-search* is updated by subtracting the search food intake of the trout as calculated in the habitat and activity selection submodel. Search food intake is limited so it cannot exceed *cell-available-search* and the value of *cell-available-search* cannot become negative. (The amount of search food depleted by a superindividual trout is multiplied by its value of *trout-superind-rep*.)

9.12 Trout memory

This submodel updates the three trout memory variables, *consumption-memory-list*, *growth-memory-list*, and *survival-memory-list*, which are lists containing the food consumption, growth rates, and survival probabilities experienced by the trout during previous time steps of the 1.0 d period that ends at *sim-time* (the end of the current time step). In doing so, it also updates the observer variable *daily-step-length-list*. The submodel is executed before habitat and activity selection each time step. The submodel is designed to work with fewer or more than the standard four time steps per day. (The habitat and activity selection submodel also affects these lists: after each trout has identified the cell and activity it will use, it records the food consumption, growth, and survival it experiences there by adding those values to the beginning of the lists; Sect. 9.13.3)

The first step in the submodel is updating *daily-step-length-list* and evaluating a variable *last-step-adjuster*; this step is done by the observer[33]. *Daily-step-length-list* contains the values of *step-length* for the preceding time steps of the 1.0 d period that ends at *sim-time*. This list is maintained by adding the value of *step-length* for the previous time step to the beginning of the list, and then removing values from the end of the list to make the sum of its values plus *step-length* as close as possible to 1.0 d. Values are removed from the end of the list one at a time until removing the next value would make the absolute value of the sum of list values plus *step-length* minus 1.0 increase instead of decrease (the total time represented by the list plus the current time step becomes further instead of closer to 1.0 d). The variable *num-steps-dropped* records the number of values removed.

The variable *last-step-adjuster* is then calculated as the value by which the last (oldest) value on *daily-step-length-list* can be multiplied to make the sum of values on *daily-step-length-list* be exactly 1.0 d. (For the habitat and activity selection submodel, it is probably not important that the trout memory lists represent exactly 1.0 d, but this adjustment allows meaningful comparison of trout growth, survival, and the fitness measure explained in Sect. 9.13.2 across days.) This variable is calculated via:

$$lastStepAdjuster = 1.0 + \left(\frac{1.0 - stepLength - \sum dailyStepLengthList}{lastStepLength}\right)$$

where *last-step-length* is the last value on *daily-step-length-list*.

The submodel's second step is to update the memory lists of each trout[34]. Each trout executes the same process for all lists, with one important exception. All trout use the values of *num-steps-dropped* and *last-step-adjuster* calculated by the observer.

For all three trout memory lists, the number of values equal to *num-steps-dropped* are deleted from the list end. Then the last values on *consumption-memory-list* and *growth-memory-list* are adjusted by multiplying them by *last-step-adjuster*. The last value on *survival-memory-list* is

[33] This part of the submodel is programmed in procedure `update-trout-memories`.

[34] This step is programmed in procedure `update-fitness-memories-with`.

adjusted by raising it to the power *last-step-adjuster* ($V = V^a$ where V is the last value on *survival-memory-list* and a is *last-step-adjuster*).

9.13 Habitat and activity selection

This submodel represents the primary adaptive behavior of the model trout: choosing which of three alternative activities to use, in which habitat cell, on each of the four time steps per day[35]. Selection of both habitat cell and activity are decided together because the choice of one depends on the other: which activity is best depends on what habitat is available, and which cell is best depends on which activity is used.

Updating habitat and activity selection throughout the circadian cycle is an innovation in inSTREAM 7 that gives its simulated fish much greater ability to adapt to changes in habitat and biological conditions than did previous individual-based fish models (Railsback et al. 2021). This approach has been tested thoroughly and shown to reproduce a variety of observed patterns in salmonid behavior (Railsback et al. 2020).

The submodel is executed by each model trout, using three steps explained in detail below:

Identify which cells (the "potential destination cells") to evaluate,

Calculate a fitness measure for each activity at each potential destination cell, and

Identify and use the combination of cell and activity with highest value of the fitness measure.

Three activity alternatives are considered:

Drift feeding ("drift") is stationary feeding, using the drift feeding submodel (Sect. 9.22). A drift-feeding trout may or may not occupy some of its cell's velocity shelter resource (Sect. 9.11): if velocity shelter is available—defined as the cell's value of *cell-available-vel-shelter* being greater than the square of the trout's length—then the trout is assumed to use velocity shelter and its swimming speed is assumed equal to the cell's velocity times the reach parameter ***reach-shelter-speed-frac***.

Search feeding ("search") is active searching for (e.g.) food on the stream bottom or in vegetation. Trout using search feeding never occupy velocity shelter and are assumed to swim at a speed equal to the cell's velocity.

Hiding ("hide") represents behaving only to reduce predation risks, and not feeding. Trout using hiding have zero food intake and are assumed to have a swimming speed of zero. Hiding can be used while occupying a hiding place (which reduces the risk of terrestrial predation; Sect. 9.18.6) or without occupying a hiding place (which, for small fish, reduces the risk of predation by other fish; Sect. 9.19.4).

Appropriate values of the reach parameter ***reach-shelter-speed-frac*** depend on the kind of velocity shelter prevalent in the reach and the extent to which it disturbs flow and creates velocity gradients trout can take advantage of. A number of studies have shown that "focal" water velocities (the velocity measured as closely as possible to the spot where a fish was drift-feeding) are related to, but less than, the depth-averaged velocity at the same location (e.g., Moyle and Baltz 1985; Naman et al. 2020). However, the relations between focal and depth-

[35] The top-level code procedure for habitat and activity selection is `select-activity-and-habitat`. The fitness measure used to rate alternatives is coded in `fitness-for`.

averaged velocities observed in such studies are not directly applicable to inSTREAM because (a) ***reach-shelter-speed-frac*** approximates the difference between a fish's swimming speed and cell average water velocity, not depth-averaged velocity at the fish, and (b) this literature does not distinguish between fish that are vs. are not actually using velocity shelter for drift feeding. For a small, hydraulically complex stream with velocity shelter due to boulders and logs, Railsback and Harvey (2001) used a value of 0.3 for ***reach-shelter-speed-frac***. For a large river with heavily embedded and rounded substrate, Railsback et al. (2005) used a value of 0.5.

9.13.1 Habitat selection radius and identification of potential destination cells

Potential destination cells are those within a radius *R*, the habitat selection radius, of the trout's current cell; the trout is assumed to have explored and be aware of habitat conditions within the radius[36].

The value of *R* has strong effects on both inSTREAM's results and its execution speed. Low values of *R* reduce the ability of model trout to adapt to habitat and competitive conditions by reducing the range of cells they select from, while too-high values can exaggerate adaptive ability. Because the number of cells evaluated as destinations increases approximately with R^2, the number of computations made by inSTREAM and its execution time increase rapidly with *R*.

Field observations of distances that trout move routinely can inform how we model *R*, even though they do not directly tell us the distances trout explore to evaluate habitat. We examined studies in which stream trout were radiotagged and located over time. Harvey et al. (1999) observed 18-24 cm Cutthroat Trout in a mid-sized stream, observing daily movement distances (maximum distance among observed locations within one day) of up to 55 m. Ovidio et al. (2002) observed large (28-43 cm) Brown Trout in a mid-sized stream. They observed "home ranges" of up to 500 m, with a mean of 50 m. They also observed more movement in warmer seasons but no relation between home ranges and trout size. Hilderbrand and Kershner (2000) observed adult Cutthroat Trout in a first-order stream; their one set of observations within a single day found movement up to 120 m. Diana et al. (2004) observed large (44-58 cm length) Brown Trout in a mid-sized river where concealment cover was scarce. They observed trout locations at random times over many days, and also observed locations of three individuals hourly for 24 h, on 36 occasions. While Diana et al. found trout to use multiple "home sites" that were up to 3.5 km apart over a year, they also estimated distances trout moved for foraging within a day. In nine trout, the median of this distance was often zero but ranged up to 120 m; the maximum distance ranged 5-650 m, but only two trout moved more than 170 m. Diana et al. (2004) found that the distance trout moved at night (presumably for foraging, from daytime concealment) was negatively correlated with stream gradient.

These observations lead to several conclusions and assumptions. First, our model of *R* relies on observations of routine habitat selection movements, not the longer migrations sometimes observed in the field, often associated with spawning. InSTREAM does not represent such migrations. Second, it may not be safe to assume lower *R* in smaller streams: narrower streams provide less habitat within the same radius. Third, *R* may not vary with trout size for adults; it is unlikely that habitat selection movement is limited by swimming ability and size. However, for small juveniles habitat exploration could very well be limited by swimming ability, so it seems

[36] The habitat selection radius is calculated in `update-logistics`, using a logistic function created in `build-logistic-functions`.

safe to assume that *R* increases rapidly with size at small lengths. Fourth, for large trout habitat conditions such as gradient and availability of concealment cover could affect how far they move among feeding and concealment sites.

R is therefore modeled as a logistic function of trout length:

R = ***trout-move-radius-max*** × logistic(*trout-length*).

The parameters are: ***trout-move-radius-max*** (cm), the maximum radius; and ***trout-move-radius-L1*** and ***trout-move-radius-L9***, the trout lengths (cm) at which *R* is 0.1 and 0.9 of ***trout-move-radius-max***. Values of 15,000 cm (150 m) for ***trout-move-radius-max***, 6 cm for ***trout-move-radius-L1***, and 20 cm for ***trout-move-radius-L9*** produce results compatible with the literature cited above. However, users are encouraged to re-evaluate these parameters using site-specific evidence or judgement of habitat exploration distances.

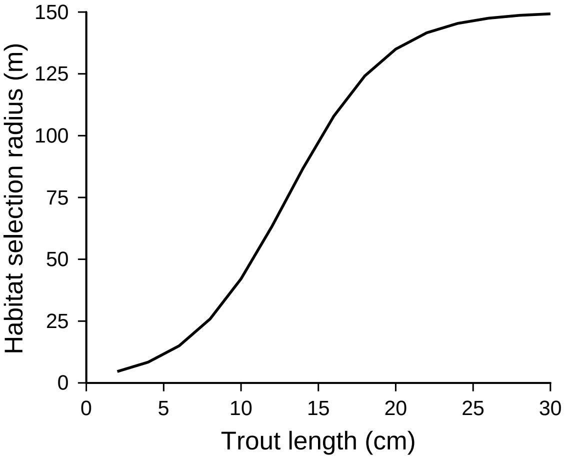


Figure 8. Logistic relation between trout length and habitat selection radius *R*, with values of 15,000, 6, and 20 cm for ***trout-move-radius-max***, ***trout-move-radius-L1***, and ***trout-move-radius-L9***. Note that the Y axis shows *R* in m, not cm.

The trout creates a list of potential destination cells in three steps.

First, the list includes those cells with centroids within a radius *R* of the centroid of the trout's current cell. This step uses the distance conventions of Sect. 2.2.9.

Second, the potential destination cells always include all cells adjacent to the trout's current cell (sharing a side or vertex), so even the smallest trout can always consider at least adjacent cells. Any adjacent cells not already on the list are added to it.

Finally, any cells on the list with depth of zero are removed from it; this precludes trout from considering or occupying dry cells. However, if all cells on the list have zero depth, then the only cell left on the list is the trout's current cell (the trout will be forced to remain in its current dry cell).

(An alternative version of inSTREAM 7 also requires that there be a way for the trout to get from its current cell to the destination cell by passing only through cells that have depth greater than zero, which excludes isolated wet cells surrounded by dry cells as destinations. This criterion could be important for sites with extensive shallow habitat or side channels.)

9.13.2 Evaluation of expected fitness

In this step, the trout calculates its fitness measure—an approximation of its expected future survival and reproductive potential—for each of the three activity alternatives at each of the potential destination cells. Evaluating the fitness measure is complex and uses many of the other submodels (Figure 9).

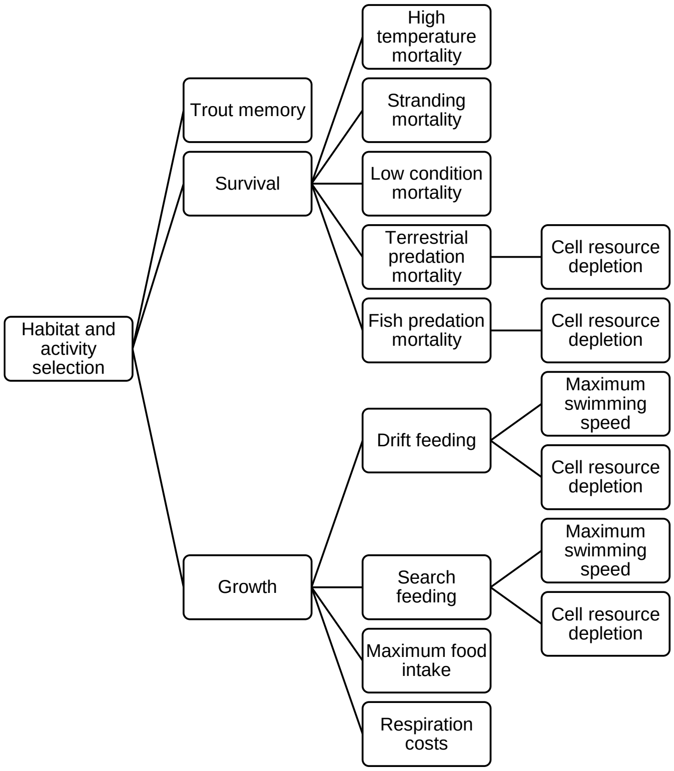


Figure 9: Submodels used in habitat and activity selection. Evaluating the fitness measure used to rate combinations of activity and habitat cell requires calculating the growth rate and survival probability that a trout would obtain, which requires many of inSTREAM's submodels.

The fitness measure can be summarized with three terms:

$F = P \times S \times R$

where *F* is the fitness measure value, *P* is the expected probability of surviving predation and all other mortality risks except starvation (Sect. 9.14) until a future time horizon, *S* is the expected probability of surviving starvation (poor condition) until the time horizon, and *R* is a reproductive success term. (See the end of this section for an exception: how *F* is determined for cells with velocity exceeding the fish's swimming ability.) *P*, *S*, and *R* are all functions of the trout, the activity, and the habitat cell for which the measure is evaluated, as explained below. All range from 0.0 to 1.0, so *F* is always between 0.0 and 1.0.

The time horizon T is a "sliding" time horizon that is always a constant number of days in the future. This number of days is specified by the trout parameter ***trout-fitness-horizon***. The time horizon controls the balance of starvation and non-starvation survival. When the horizon is low (<40 d, in simulation experiments), trout give less emphasis to starvation in their decisions because the risk of starvation over a short time period is low (except when condition is extremely low; Sect. 9.17). Trout therefore sacrifice condition to avoid predation risk (*trout-condition* falls below 1.0), and as their condition decreases the trout suffer poor condition mortality. At higher values of the time horizon, the poor condition survival submodel (Sect. 9.17) produces high risk of starvation unless trout feed enough to keep their condition near 1.0. This produces a good balance between feeding and avoiding risk and, therefore, higher trout abundance and biomass than do low time horizons. A simulation experiment examining sensitivity to ***trout-fitness-horizon*** found highest values of trout abundance and biomass over a range of values ~40–90 d (Figure 10). We recommend a value of 60 d. (The value of ***trout-fitness-horizon*** does not affect model execution time.)

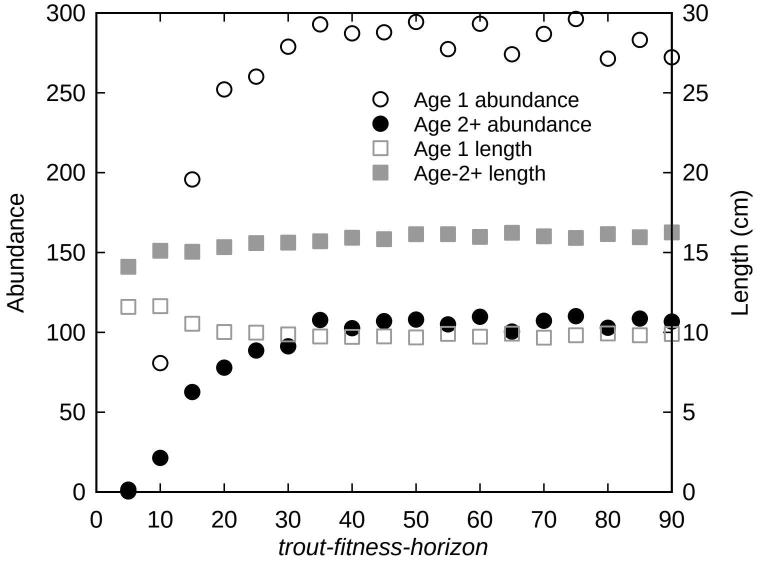


Figure 10: Sensitivity of predicted trout abundance and mean length to the parameter ***trout-fitness-horizon***. Mean results over the last 8 years of 10-year simulations at the "RESTORED" site of Railsback et al. (2021).

The fitness measure uses the simplifying approximation that the growth and survival conditions experienced over the 1.0-day period ending with the current time step would persist until the time horizon. The daily growth rate and survival probability are determined by combining the conditions experienced by the trout in previous time steps with what it would experience in the current time step for the combination of activity and cell under consideration. The model trout maintain a "memory" of the non-starvation survival probabilities (P_i, for i = 1 to n) and growth rates (G_i) they experienced with the habitat cell and activity they chose in each of the n phases preceding the current one (Sect. 9.12). (For a normal circadian cycle with four time steps per day, n is 3.) For example, when a trout is selecting habitat and activity at the start of a daytime phase, P_i and G_i are the non-starvation survival probabilities and growth rates experienced in the previous dawn, night, and dusk phases. These memories are combined with the survival and growth rates for the alternatives being considered for the current phase to determine the overall rates for a full circadian cycle, adjusting for the duration of each phase.

The 1.0-day growth rate used to evaluate the fitness measure, G_D, is calculated as:

$$G_D = \left(\sum_{i=1}^{n} G_i\, d_i\right) + (G_c d_C)$$

where G_i are the growth rates (g/d) experienced in the *n* previous time steps of a 1.0 day period that ends at the end of the current time step, d_i are the durations (d) of each such time step, G_C is the growth rate (g/d) for the current time step, and d_C is the duration (d) of the current time step (equivalent to the observer variable *step-length*). When selecting activity and habitat, the first term of this equation is calculated only once per trout per time step, while the second term and G_D are evaluated for each combination of activity and cell being considered.

Similarly, the 1.0-day non-starvation survival probability used in the fitness measure, P_D, is:

$$P_D = \left(\prod_{i=1}^{n} (P_i)^{d_i}\right) (P_c)^{dc}$$

where P_i are the daily non-starvation survival probabilities experienced by the trout in the *n* previous time steps and P_C is the non-starvation survival probability for the current time step. Again, the first term of the equation is constant over all the combinations of activity and cell being evaluated at one time step.

The fitness measure's term for expected probability of surviving predation and other non-starvation risk until the time horizon *T* days in the future uses the approximation that the survival probability for the current full day would persist to the time horizon. Therefore, the term is: $P = P_D{}^T$.

The term for expected probability of surviving starvation until the time horizon is more complicated because starvation is a function of the trout's condition (*trout-condition*; Sect. 4.1.4) and how condition changes from the current time step to the time horizon. The expected probability of surviving starvation until the time horizon is calculated by projecting the trajectory of *trout-condition* (abbreviated here as *K*) from the current time until *T*. This projection assumes that the daily growth rate G_D will persist until the time horizon. Under this assumption and the restrictions that *K* cannot be above 1.0 (weight gain when *K* = 1.0 produces growth in length that keeps *K* at 1.0) or below 0.0, there are seven possible kinds of trajectory in *K* over a time horizon, depending on its initial value and whether G_D is positive or negative (left panel, Figure 11). If G_D is positive, *K* increases linearly until it reaches 1.0 or the time horizon is reached; if G_D is negative, *K* decreases linearly until it reaches 0.0 or the time horizon is reached. With our linear model of low condition mortality, the projected daily probabilities S_t of surviving this mortality follow similar trajectories (right panel, Figure 11).

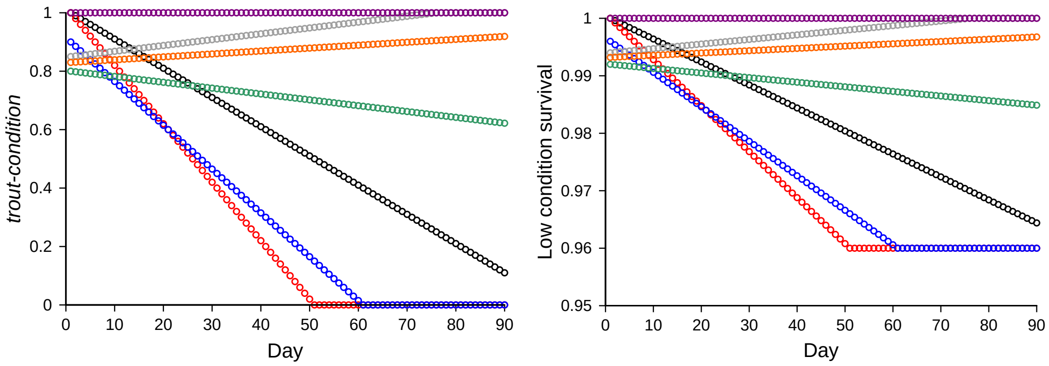


Figure 11: Possible trajectories in trout-condition (left) and the resulting trajectories in daily survival probability for low condition mortality (right), over a time horizon (***trout-fitness-horizon***) of 90 d.

The probability S of surviving low condition until the time horizon can be calculated exactly by calculating the projected daily values of K_t and S_t and multiplying the values of S_t together (i.e., multiplying together all the dots on a trajectory in the right panel of Figure 11). However, for computational efficiency we approximate S as $S_m{}^T$, where S_m is the first moment of the survival trajectory: the area underneath a trajectory in the right panel of Figure 11 divided by T (which is 90 d in Figure 11). In a comparison of 100,000 randomly-generated trajectories such as those in Figure 11, this first moment method produced results very close to the exact calculation, with a maximum difference in predicted survival between the methods of 0.016.

We use an algorithm to project the future trajectory of K and S_t, and the area under the trajectory of S_t, from the current condition (K_1) and G_D. In this algorithm, the value of survival for any value of K is calculated using the low condition mortality function of Sect. 9.17[37].

If K_1 is 1.0 and G_D is zero or positive, then K and S_t are 1.0 for all days to the time horizon. In this case, $S = 1.0$.

If G_D is near zero, then K and S_t are equal to K_1 and S_1 for all days to the time horizon. We define "near zero" as G_D having an absolute value less than 0.0001 g/d, which produces negligible growth over a 90-day time horizon even for a newly emerged trout. In this case, $S = S_1{}^T$.

For the remaining steps, convert G_D to a daily rate of change in condition, ΔK. The value of ΔK is G_D divided the weight of a healthy trout of the current length (weight of a trout with the length *trout-length* and K of 1.0).

If G_D is negative, calculate the time (in days from the current time) at which K reaches zero ($t_{K=0}$): $t_{K=0} = K_1/(-1 \times \Delta K)$.

If $t_{K=0}$ is less than or equal to T, K reaches zero before the time horizon. The area under the survival trajectory is equal to the sum of the areas of (a) the trapezoid defined by survival decreasing from S_1 to the survival probability when K is 0.0 ($S_{K=0}$) over the time $t_{K=0}$, and (b) the rectangle with height equal to $S_{K=0}$ and width equal to $T - t_{K=0}$.

[37] This algorithm is implemented in the code procedure `mean-condition-survival-with`.

If $t_{K=0}$ is greater than T, K stays above zero until the time horizon. The condition at T, K_T, is calculated as: $K_T = K_1 + (T \times \Delta K)$. The area under the survival trajectory is then equal to the area of the trapezoid defined by survival decreasing from S_1 to the survival probability at K_T over the time T.

If G_D is positive, calculate the time at which K reaches 1.0 ($t_{K=1}$): $t_{K=1} = (1.0 - K_1)/\Delta K$. (The case in which K_1 is 1.0 is handled at step 1.)

If $t_{K=1}$ is less than or equal to T, K reaches 1.0 before the time horizon. The area under the survival trajectory is therefore equal to the sum of the areas of (a) the trapezoid defined by survival increasing from S_1 to 1.0 over the time $t_{K=1}$, and (b) the rectangle with height equal to 1.0 and width equal to $T - t_{K=1}$.

If $t_{K=1}$ is greater than T, K stays below 1.0 until the time horizon. The condition at T, K_T, is calculated as: $K_T = K_1 + (T \times \Delta K)$and the area under the survival trajectory is the area of the trapezoid defined by survival increasing from S_1 to the survival probability at K_T over the time T.

(In this algorithm, with the linear model of low condition mortality of Sect. 9.17, daily survival is well above zero when a trout's weight is projected to zero. This assumption is certainly unrealistic but still useful because it produces behavior causing model trout to keep their condition high.)

The fitness measure's reproduction term R represents the benefits of trout size to future fitness. InSTREAM 7 retains the reproduction term of most previous versions (and Railsback et al. 1999), which uses reaching a target length as the measure of reproductive benefit. Previous versions of inSTREAM (through version 7.2) used the minimum length for spawning (***trout-spawn-min-length***, Sect. 9.27.1) as the target length, but that approach does not always produce realistic length distributions of adult trout in inSTREAM 7. Starting with version 7.3, the target length is a parameter ***trout-fitness-length*** (cm), which represents the length at which further growth does not provide substantially increased fitness. The term R is evaluated by first projecting the trout's expected length at the time horizon L_T: if K_T is 1.0, then L_T is the length of a healthy fish with weight W_T, as determined from the methods in Sect. 9.20; otherwise, the trout is not expected to grow in length to the time horizon so L_T is equal to *trout-length*. The value of R is then 1.0 if L_T equals or exceeds ***trout-fitness-length***, and otherwise is L_T / ***trout-fitness-length***.

Trout-fitness-length can be given a value approximating the size of a large adult. Except under poor growth conditions, the lengths of the largest model trout will exceed this parameter's value. Its value can be adjusted to produce realistic lengths of older adults, but ***trout-fitness-length*** should at least equal ***trout-spawn-min-length***. This parameter can also affect simulated trout abundance, so its value should be set before or as part of calibration (Sect. 24).

An exception to the above methods for calculating F is made for cells where the trout would have to swim at a speed higher than its maximum sustainable speed. This exception is the only method in inSTREAM 7 that essentially restricts trout from using such cells. The exception is a set of simple rules: if the cell's velocity exceeds the trout's maximum sustainable swimming speed in the cell (Sect. 9.26), then F for the "search" activity is set to 0.0; and if the speed at which a drift-feeding trout would have to swim in the cell (which depends on whether sufficient velocity shelter for the trout is available, which depends on trout length; sects. 9.11, 9.13) exceeds the trout's maximum sustainable swimming speed, then F for the "drift" activity is set to

0.0. Cell velocity does not affect fitness for the "hide" activity because hiding trout are assumed not exposed to the water column velocity. (Previous versions of inSTREAM used a "high velocity" mortality source to discourage fish from using cells where they would exceed their maximum swimming speed. The inSTREAM 7 approach described here is simpler and requires less computation.)

It is possible for simulated trout to have access to no cells where they could swim at less than their maximum sustainable speed. In that situation, trout select the hiding activity because it always offers $F > 0.0$. This behavior allows trout to hide through temporary high-flow events. However, the trout most likely to be in this situation are newly emerged fry that find no cells with velocity below their maximum swimming speeds near their natal cell. Such trout will select the hiding activity but also select the shallowest available cell, because lower depth reduces their risk of predation by fish (Sect. 9.19.2). Seeking the lowest depth over several time steps allows such trout to move toward shallower cells where velocities are more likely to allow feeding.

9.13.3 Selection and implementation of the best alternative

Once the trout has evaluated the fitness measure for all combinations of cell and activity, it identifies the combination offering highest fitness. (In the unlikely case of multiple combinations offering the same fitness, one is selected randomly.) The trout then occupies the cell and depletes the cell resources for the selected activity (Sect. 9.11). The trout then also updates its memory lists (Sect. 9.12): the food consumption it experiences with the selected cell and activity (g food consumed during the time step; zero if the activity is hiding) is added as the first item on *consumption-memory-list*, the growth rate it experiences (g/d) is added to the beginning of *growth-memory-list*, and the survival probability it experiences is added to the beginning of *survival-memory-list*.

9.14 Trout survival

The survival submodel simulates whether and why trout survive or die. Each trout executes it each time step[38].

The mortality submodels are used not only by this trout survival submodel but also in the habitat and activity selection submodel (Sect. 9.13); the predation and condition mortality submodels have especially strong effects on habitat and activity selection, which then affects growth and survival. Users should therefore be aware that changes to these submodels and their parameters will affect how model trout behave as well as how many of them die of what causes. For example, changing parameter values to make survival of predation by terrestrial animals higher at shallow depths may increase trout survival but it also allows trout to adjust their tradeoff between growth and risk so they make more use of shallow habitat that, while less risky, is still risky; as a consequence of this behavior, the parameter change may result in less-than-expected change in survival but an increase in growth.

The mortality submodels make extensive use of logistic functions, which have several useful characteristics. Their shape can represent the way many variables affecting survival are benign over a wide range of values, then have sharply increasing effects, and then can cause near-complete mortality over a wide range of extreme values. Logistic functions can be fit to observations of survival using logistic regression. They also produce survival probabilities that

[38] The submodel is coded in procedure `survive`. Supporting code that updates the logistic functions used in survival submodels is in `update-logistics`.

approach 0.0 asymptotically as conditions become worse, allowing model fish to make good decisions even when all the alternatives are bad. Conversely, however, logistic survival probabilities are never 100% so some mortality occurs randomly even under benign conditions.

The survival submodel executes two simple steps for each of the trout mortality sources. The order in which the different mortality sources are simulated is specified in the model schedule (Sect. 5).

The first step is to calculate the probability of the trout surviving the mortality source for the current time step. The daily survival probability is obtained from the relevant mortality submodel (sects. 9.15 through 9.19), and then adjusted for time step length: $S_{ts} = S_d^{step\text{-}length}$ where S_{ts} is the probability of surviving the time step and S_d is the daily survival probability from the mortality submodel.

The second step is to execute a random Bernoulli trial to determine whether the trout survives the mortality source. A random floating point number between 0.0 and 1.0 is drawn and the trout dies if the random number is greater than the survival probability S_{ts}. If the trout dies, it is immediately removed from the model and executes no further actions, so it cannot possibly die of a subsequent kind of mortality. Consequently, the number of trout dying of each mortality source depends (though, normally, relatively little) on the order in which the sources are scheduled in the survival submodel.

If the trout does not die of any of the mortality sources, it survives and is unaffected by the submodel.

(Previous versions of inSTREAM included a “high velocity” mortality source, which is not included in inSTREAM 7. That mortality source decreased survival for fish using cells where their swimming speed was high in comparison to their maximum sustainable swimming speed; it was used to discourage trout from selecting such cells. InSTREAM 7 uses a simpler method—assuming zero fitness if the maximum swimming speed is exceeded, Sect. 9.13.2—for the same purpose.)

9.15 High temperature trout mortality

This mortality source represents the acute breakdown of physiological processes at high temperatures[39]. It does not represent chronic effects of high temperatures on bioenergetics, which are instead captured in the growth and low condition mortality submodels. High temperature mortality also does not represent the effect of disease even though fish are probably more susceptible to disease at high temperatures. Instead, disease is modeled as part of poor condition mortality; a fish able to maintain its weight at sublethal temperatures is assumed to remain healthy.

Temporal resolution can be important to modeling temperature mortality: fish may be able to survive temperatures for short periods that would cause high mortality over longer times. InSTREAM 7 can use temperature input of daily values or separate values for each time step (e.g., by using hourly data; Sect. 8), so parameters could be adjusted to reflect the resolution of input. For example, if input is daily mean temperature, then parameter values could be chosen to reflect the assumption that daily peak temperatures are higher than the input (although the relative importance of mean v. maximum temperature is not clear; Dickerson and Vinyard 1999,

[39] High temperature survival is programmed in the procedure `survival-high-temperature-for`.

Hokanson et al. 1977). The temperature mortality parameters can be re-evaluated for sites with particularly high or low diurnal temperature variations.

High temperature mortality has been addressed by numerous laboratory studies, but models of this mortality remain variable and uncertain because susceptibility varies with laboratory conditions and techniques and the endpoints used to define mortality, between laboratory and field conditions, and among individuals. Available literature (e.g., Behnke 1992; Myrick and Cech 2004) indicates that any differences in measured lethal temperatures among trout species are not clearly distinguishable from uncertainty and variability in the measurements. Myrick (1998) found approximately 60 percent survival of well-fed golden trout (*Oncorhynchus mykiss*) juveniles over a 30-d period at a constant 24°, equivalent to a daily survival of 0.98. Dickerson and Vinyard (1999) measured survival of Lahontan Cutthroat Trout (*O. clarki*) for 7 d at high temperatures, finding zero survival at 28°, 40 percent survival at 26° (equivalent to daily survival of 0.88), and 100 percent survival at 24°. This literature indicates that high temperature mortality can be modeled well as a logistic function.

The daily survival probability for high temperature mortality is therefore modeled as a logistic function of the reach variable *temperature*, with its shape defined by the trout parameters ***mort-high-temp-T1*** and ***mort-high-temp-T9***. The parameters in Table 8 (illustrated in Figure 12) appear suitable for sites with relatively low diurnal variation in temperature; they produce daily survival of 0.98 at 24°, 0.88 at 26°, and < 0.5 at 28°.

Table 8: Parameters and values for high temperature mortality

Parameter name	Definition	Basis	Standard value
mort-high-temp-T1	Temperature at which daily survival probability for high temperature mortality is 0.1	*Oncorhynchus spp.* literature cited above.	30
mort-high-temp-T9	Temperature at which high temperature survival is 0.9		25.8

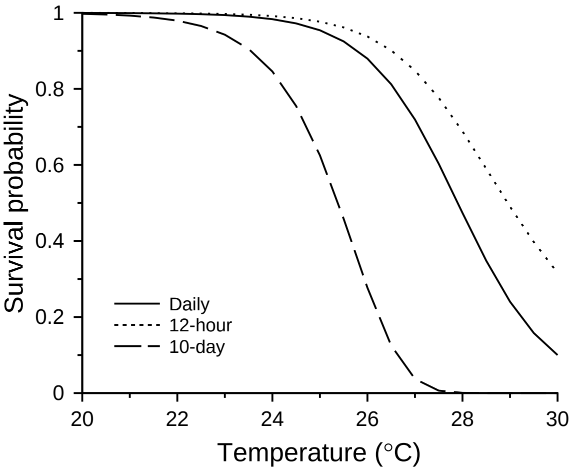


Figure 12: Survival submodel for high temperature, using the parameter values in Table 8, showing daily survival and the corresponding probability of surviving for 12 hours and 10 days.

9.16 Stranding trout mortality

The stranding mortality submodel represents the risk of being trapped in a cell that is not submerged at the current flow; stranding can be a concern when flows change rapidly[40]. Previous versions of inSTREAM represented survival of stranding as a logistic function that decreased as the ratio of cell depth to fish length decreased, reflecting (a) that cell beds are typically sloped so they gradually dry as flow approaches and declines below that at which simulated cell depth reaches zero, (b) the need of larger fish for greater depth to avoid stranding, and (c) a continually increasing risk of predation (due to, especially, wading birds and snakes) as depth decreases. However, that formulation had two problems: it mixed mortality due to terrestrial predators with stranding, and it could produce substantial levels of stranding mortality even under conditions when actual stranding would be assumed negligible.

To avoid these problems, stranding mortality is modeled in inSTREAM 7 using a simple, binary approach that assumes stranding mortality happens only when depth is zero: if *cell-depth* > 0.0, then survival is 1.0, and if *cell-depth* = 0.0 then survival of stranding is equal to the trout parameter ***mort-strand-survival-when-dry***. This can happen only when a trout occupies a cell that becomes dry when flow decreases, because trout are not allowed to move to a dry cell (Sect. 9.13.1). This parameter is given a standard value of 0.5 to reflect that cells where mean depth recently dropped to zero may still include some water. Small trout are more vulnerable to this mortality source because the distance they can move to avoid stranding is less than for large trout (Sect. 9.13.1).

9.17 Low condition trout mortality

Trout in poor condition (with low values of *trout-condition*, weight in relation to length) are at risk of starvation, disease, and increased vulnerability to predators[41]. (In addition, low condition makes trout less able to grow and reproduce, additional negative effects on fitness.) These risks are combined in the low condition survival probability. This mortality submodel is especially important because of its role in the habitat and activity selection submodel (Sect. 9.13.2): model trout make decisions trading off growth and predation risk by balancing predation survival against low condition survival. Therefore, seemingly small changes to this submodel can strongly affect trout behavior and simulated population dynamics even if they do not produce widespread low condition mortality.

Simpkins et al. (2003a, b) studied starvation mortality in large juvenile trout, finding:

Trout can survive for long periods (over 147 d, in some cases) with no food,

Survival is lower at higher swimming activity and temperature (which both increase metabolism),

Relative weight (equivalent to *trout-condition*) decreases linearly over time during starvation, but

[40] Stranding survival is programmed in the procedure `survival-stranding-for`.

[41] Low condition survival is in the code procedure `survival-condition-for`.

Mortality is predicted better by an index of lipid content than by *trout-condition*; one reason is that lipids are replaced by water as energy stores decrease.

Modeling how body lipids are depleted and replaced by water would add considerable complexity and uncertainty to inSTREAM, as these processes are not well understood. Instead, low condition survival probability is represented as simple function of *trout-condition* that produces reasonable probabilities of surviving over several days and weeks.

InSTREAM 7 models low condition mortality differently than previous versions. Previous versions of inSTREAM represented survival of low condition as a logistic function of *trout-condition* that produced high (but not 100%) survival over high values of *trout-condition* but rapidly decreasing survival at *trout-condition* values below about 0.7. The logistic function's low gradient of survival with *trout-condition* near condition = 1.0 gave trout little incentive to keep condition high, which can prevent individuals from growing in length (Sect. 9.20) or spawning (Sect. 9.27.1). This limitation had much stronger effects in inSTREAM 7, which gives model trout more precise ability to balance starvation and other risks. To prevent this problem and to avoid the possibility of low condition mortality when *trout-condition* is 1.0, inSTREAM 7 uses a linear relation between survival probability and *trout-condition*. This relation is defined by the parameter ***mort-condition-S-at-K5***, the probability of surviving low condition mortality for one day when *trout-condition* is equal to 0.5. The submodel represents condition survival probability as a straight line defined by two points: the value of this parameter when condition is 0.5 and survival of 1.0 when condition is 1.0. The linear relation between condition and daily survival produces an exponential relation between condition and the probability of surviving over a future period (Figure 13).

The value of ***mort-condition-S-at-K5*** controls the emphasis given to maintaining high condition versus avoiding predation risk in trout behavior. Our explorations of the submodel and its effects on habitat and activity selection behavior indicate that values of 0.98 produce reasonable tradeoffs (Figure 13): model trout will still reduce their condition to avoid high risk, but will otherwise feed enough to keep condition near 1.0 and continue growing. Higher values (e.g., 0.99) make model trout even more willing to avoid risk by maintaining lower condition but, therefore, less able to grow or spawn.

The assumption that daily survival is 0.98 when *trout-condition* is 0.5 may seem more generous than it is. Low condition is a unique mortality source in that fish can never increase their survival probability immediately by selecting different habitat or activity. Fish in poor condition have a strong incentive to select rapid growth so their condition increases; however, recovering condition requires high growth for a number of days. Even apparently high daily survival probabilities for this mortality source result in a substantial risk of dying before normal weight can be regained. For example, an adult trout that has its value of *trout-condition* reduced to 0.8 by spawning (Sect. 9.28) and regains its condition at a steady rate of 0.5% per day has a 83% probability of surviving low condition mortality over the 45 days needed to regain *trout-condition* of 1.0 (and is typically also exposed to an equivalent level of predation risk).

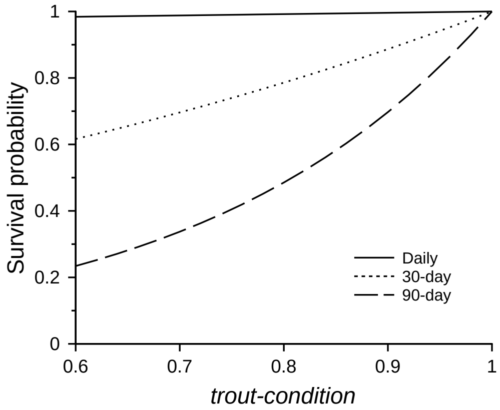


Figure 13: Probability of surviving low condition mortality for 1, 30, and 90 days, with ***mort-condition-S-at-K5*** = 0.98. The 30- and 90-day curves assumed that condition is constant over those periods at the X axis value.

9.18 Terrestrial predation trout mortality

Predation by terrestrial animals is a dominant source of mortality to trout, especially adults (Alexander 1979, Harvey and Stewart 1991, Metcalfe et al. 1999, Quinn and Buck 2001, Valdimarsson et al. 1997)[42]. The terrestrial predation formulation represents predation by a mix of such predators as otters, raccoons, snakes, herons, mergansers, kingfishers, and dippers. Common characteristics of terrestrial predators that affect the survival probability function include their: large size compared to trout, limited swimming ability, dependence on locating fish from above water, and warm-blooded physiology. These characteristics vary among predators, but lead to these generalizations about terrestrial predation:

Risk does not necessarily decrease with trout size,

Risk persists year-round because warm-blooded predators feed as much or more in winter (except those that hibernate or migrate),

Trout are more at risk when more visible from the air, and

Trout face higher risk at higher light levels (during daytime) and when feeding versus hiding.

(Ice cover in winter can strongly limit terrestrial predation, but inSTREAM 7 does not represent this process; Sect. 1.2.)

The formulation assumes a minimum survival probability that applies when fish are most vulnerable to terrestrial predation, and a number of “survival increase functions” that can increase the probability of survival above this minimum. Survival increase functions have values (termed “survival increase”) between zero and one, with higher values for greater protection from predation. InSTREAM 7, unlike previous versions, assumes that survival increase functions interact: several survival increase functions that have moderate benefits can combine to reduce risk by more than each function does by itself. The interaction among survival increase functions is modeled by treating each function’s survival increase as a probability and calculating the joint probability of surviving all of them. This assumption is expressed mathematically as:

[42] Survival of terrestrial predation is programmed in the procedure `survival-terr-pred-for`.

$$S_T = reachTerrPredMin + \left((1.0 - reachTerrPredMin) \times \left(1.0 - \prod_{i=1}^{i=n} (1.0 - I_i) \right) \right)$$

where S_T is the daily probability of surviving terrestrial predation, ***reach-terr-pred-min*** is a reach parameter representing the daily survival probability in the riskiest combination of habitat and activity, and I_i represents the n survival increase functions, described below. The value of ***reach-terr-pred-min*** is typically estimated by calibration of model results to trout abundance observations. Unlike almost all other survival parameters, ***reach-terr-pred-min*** is a reach parameter instead of a trout parameter, so that it can be varied among reaches with different levels of predation but not among trout species. As a consequence, any differences among species in predation vulnerability must be represented via the survival increase function parameters, which can differ among species.

InSTREAM results can be quite sensitive to terrestrial predation parameter values (Sect. 26.2). This sensitivity is not surprising, considering that terrestrial predation often causes most adult mortality. The parameters that define the survival increase functions for depth, velocity, and distance to escape cover have especially strong effects. If these parameters are set in such a way that the survival increase function is very close to 1.0 in several or many cells, then trout occupying those cells can be almost immune to mortality. For example, if the "small stream" parameters for depth illustrated in Figure 26 were used in a large river with many cells having depth greater than 200 cm, then trout in these cells would have very low terrestrial predation risk and could live for many years. Changing the parameters for the depth function could greatly change the amount of habitat with very low predation risk. (In reality, rivers with extensive deep water also likely have predators that can be effective in deep water.) Likewise, if the velocity and distance to escape cover survival increase functions are very steep and near 1.0 for some cells, some parts of the simulated habitat can be nearly risk-free, producing higher populations of adult trout.

The following survival increase functions are included in the terrestrial predation mortality submodel. Suggested parameter values are provided at the end of the section (Table 9). The design of the terrestrial predation submodel (and its software) make it easy to add new survival increase functions. Existing functions can be essentially turned off (so they have negligible effect on survival) by setting their parameter values to yield function values near zero, e.g., by setting the parameter for survival increase of 0.1 to -100 and survival increase of 0.9 to -120 for a variable that has only positive values.

9.18.1 Trout length

Small fish are assumed less vulnerable to terrestrial predation because they are less visible (Power 1987), less desirable, and possibly more difficult to capture, than larger fish. For example, Hodgens et al. (2004) reported that 48 trout eaten by heron ranged 3-38 cm in length, but 85% were between 10 and 28 cm; Collis et al. (2001) found that cormorants and terns feeding on outmigrating salmon and steelhead selected for the largest species; Källo et al. (2023) found cormorant predation risk for migrating trout to depend significantly on trout length, with highest risk at 35-43 cm. On the other hand, dippers are terrestrial predators that select trout fry (Thut 1970), so even very small fish can face risk from terrestrial predation. Therefore, survival of terrestrial predation is often best assumed to decrease with fish length, with no size enjoying complete protection (Figure 14).

Several factors influence the determination of parameter values for this function. One is the kind of predators believed most important and their prey size-selectivity.

Another consideration for parameterizing fish predation survival is the availability and kind of hiding cover in locations likely to be occupied by juvenile trout. While the model explicitly represents hiding or concealment cover for adults (Sect. 9.18.6), the trout length function addressed here is the way that inSTREAM7 (indirectly) represents hiding cover for small trout. If habitat that provides abundant hiding places for small trout is widespread in shallow habitat where small trout are likely to be, then this length function should be parameterized to provide relative safety to fish small enough to use that cover and not to fish too large to use it. For example, if small boulders and large cobble provide abundant crevices for trout up to ~8 cm length, then this function should be parameterized so that trout are relatively safe at lengths below, but not above, 8 cm. On the other hand, if the only cover for small fish is unembedded small cobble, then only very small trout (e.g., < 4 cm) can be assumed safe because of it.

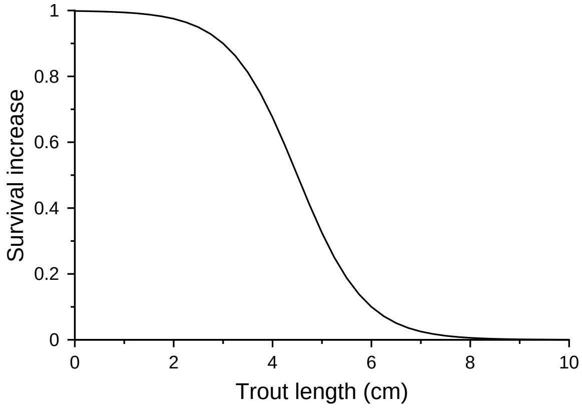


Figure 14: Survival increase function for effect of trout length on terrestrial predation mortality, using the parameter values from Table 9. Trout fry are typically born with a length >2.5 cm (Sect. 9.38), so none have a function value above about 0.95.

9.18.2 Depth

Fish are more vulnerable to terrestrial predators when in shallow water (Harvey and Stewart 1991), where they are easier for predators to locate and catch. The depth survival increase function is an increasing logistic curve: survival increases as *cell-depth* increases (Figure 15). This function represents the direct effect of depth on predation risk, not the indirect effect of lower light levels caused by depth; light is treated as a separate function (Sect. 9.18.4). (Note that inSTREAM 7, unlike previous versions, does not include high risk of terrestrial predation when fish are in near-zero depths as part of stranding mortality; Sect. 9.16.)

A variety of literature supports our parameter values. Power (1987) indicates that predation by birds is low at depths above 20 cm, and Hodgens et al. (2004) report that 85% of successful strikes by herons were at depths less than 20 cm but some were at depths up to 50 cm. However, predators that are larger or better swimmers (mergansers, otters) are effective at greater depths, especially in clear water. Our own studies of large juveniles in pool habitat (Harvey and White 2017) indicated that they became increasingly reluctant (required higher food reward) to use depths decreasing from 50 to 30 cm, and never would feed in depths as low as 20 cm.

Appropriate values for the depth survival increase function parameters can differ among sites. Parameters useful in relatively small streams of coastal California (Railsback and Harvey 2001) provide high relative survival in depths > 1 m. However, these parameters were not useful for the much larger Green River in Utah, where depths can be several meters and otters are prevalent; separate parameters were developed for the Green River site (Railsback et al. 2006). Figure 15 illustrates parameter values for small streams and large rivers (Table 9).

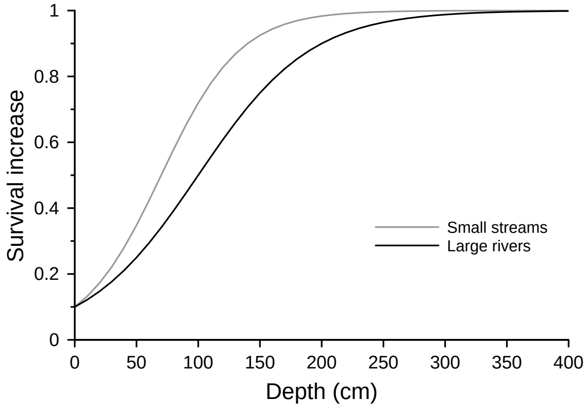


Figure 15: Survival increase function for depth effect on terrestrial predation mortality, using the parameter values from Table 9 for small streams and large rivers.

9.18.3 Velocity

Water velocity is assumed to have an effect similar to that of depth, making trout harder for predators to see and capture. Predators with limited swimming ability (e.g., snakes) may be unable to maneuver effectively at high velocities. Predators such as birds that depend on seeing trout from above the water may be most affected by how velocity interacts with other factors such as substrate type and water surface slope to make the water surface uneven and light refraction high and variable. High velocities may have less effect on predator effectiveness in larger rivers where there is more maneuvering room and, on average, smoother water surfaces. We provide separate parameter values for this logistic function, representing small streams and large rivers (Figure 16).

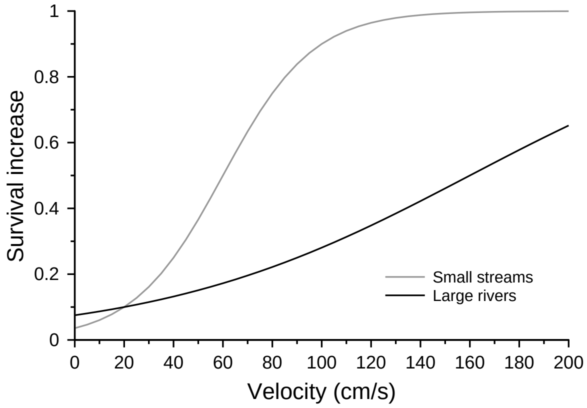


Figure 16: Survival increase function for effect of water velocity on terrestrial predation mortality, using the parameter values from Table 9 for small streams and large rivers.

9.18.4 Light

This function represents how irradiance, including the effect of turbidity, affects predation survival. Previous versions of inSTREAM had a function for turbidity and (in some versions) a separate method representing how survival probability differs between day and night. This function is driven by cell irradiance (*cell-light*) and therefore combines the effect of both light availability and turbidity, even though these two variables could affect predation in somewhat different ways.

Turbidity obviously makes fish less visible to terrestrial predators, but not all terrestrial predators rely entirely on vision. Turbidity has been observed to have negligible or even positive effects on predation risk (Gremillet et al. 2012; Hostetter et al. 2012). We assume a turbidity effect on terrestrial predation risk that parallels the observed effect of turbidity on the ability of fish to detect prey (Sect. 9.22.1), which shows the ability to detect drifting invertebrates declining toward zero at 40 NTUs. Fish are likely more visible than invertebrates because of their size, but terrestrial predators must observe prey through greater distances than must fish predators.

Some literature addresses differences between night and day light conditions on survival. Our studies in small streams of coastal California (Harvey and Nakamoto 2013) found 36% of predator attacks on adult trout occurring at night, with birds (including owls, at night) and mammals being the most common predators. From a review of literature addressing this issue, Metcalfe et al. (1999) reached the general conclusion that otter and mink are primarily nocturnal but most predation is by birds, which are usually daytime predators. Metcalfe et al. (1999) estimated that predation risk during foraging by river fish is 150 times greater during daytime than at night, in a system where 90% of mortality was due to birds and fish were active and exposed to predation only 10% of the daytime and 90% of the nighttime. However, at sites where mammals cause a higher percent of the terrestrial predation mortality, the difference between night and daytime risks can be assumed lower. These considerations led us to standard light parameter values that assume a substantial reduction in risk in twilight and about an 80% increase in survival probability at night (Table 10; Figure 17).

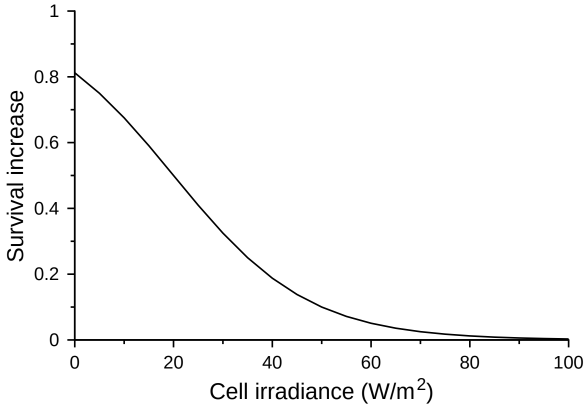


Figure 17: Terrestrial predation survival increase function for cell irradiance, as affected by surface light, turbidity, and depth. Irradiance is typically > 100 during daytime, around 20 during twilight, and < 1.0 at night.

9.18.5 Distance to escape cover

This function represents the benefits of escape cover, as quantified by the cell variable *cell-escape-dist*. The function simply assumes that the closer a trout is to a place where it can temporarily evade or avoid detection by a predator, the lower its risk of being killed. The value of escape cover is one habitat function that can occur at a spatial scale different from the cell size typically used in inSTREAM; escape cover need not be within a trout's cell to provide some protection. Sect. 22.2 provides guidance on estimating values of *cell-escape-dist*.

The benefit of escape cover to feeding fish is represented with a survival increase function that increases as distance to escape cover decreases. The value of *cell-escape-dist* can range from near zero, for cells where a bottom of boulders or vegetation provides almost continuous cover, to many meters for cells lacking bottom cover and far from the banks (e.g., extensive gravel bars). Very short distances to escape cover (< 100 cm) may provide nearly complete protection from some predators, but do not protect fish from predators that strike very quickly (e.g., some birds) or that could be able to extract trout from cover (e.g., otters). Cover several meters away may have some value for escaping from terrestrial predators that have been detected. Harvey and White (2017) observed, in a controlled experiment, that ~12-cm Steelhead Trout behaved as if escape cover reduced their fear while feeding only if within approximately one meter. Therefore, the effect of distance to escape cover is modeled as a decreasing logistic function of *cell-escape-dist* that never provides complete protection and decreases sharply to provide relatively little protection at more than 100 cm (Figure 18).

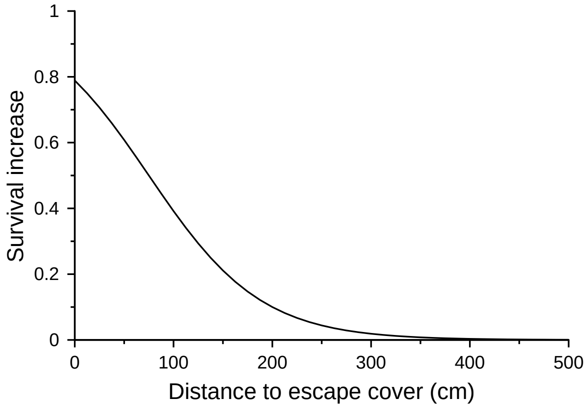


Figure 18: Terrestrial predation survival increase function for distance to escape cover.

9.18.6 Hiding activity and use of hiding places

This function represents the reduction in predation risk for fish that select the hiding activity instead of feeding. For terrestrial predation, we assume that hiding activity has no survival benefit unless trout occupy hiding places, which are defined as places suitable for adult trout hiding. Therefore, this function represents the value of hiding cover for larger trout, while the definition of "larger" trout depends on parameter values. (See the trout length function at Sect. 9.18.1 concerning hiding by small trout.)

The literature and our own research (e.g., Harvey and White 2016) indicates that the willingness of real trout to use hiding cover and, presumably, the value of cover in reducing risk can be a complex function of cover characteristics, the trout's size, and the number and characteristics of the other fish using the cover. Trout appear more willing to compete for hiding space than to share it. Harvey and White (2016) concluded from a laboratory study that the willingness of trout to use cover, in particular to share it with other trout, depends strongly on visual isolation among the individual fish. Trout seemed unwilling to use relatively large areas (much larger than the square of their length) that were already occupied, but salmonids have been observed hiding in high densities in cover complex enough to conceal them from each other, e.g., juvenile salmon hiding in leaf packs. Similar competition has been observed by Armstrong and Griffiths (2001) and Harwood et al. (2002). For this reason, inSTREAM 7 differs from previous versions by representing hiding cover as an integer number of separate locations in a habitat cell where a fish could hide in isolation.

Whether a model trout obtains the benefits of hiding in a particular cell therefore depends on whether at least one hiding place is available, i.e., when the cell variable *cell-available-hiding-places* is at least one (or, for superindividuals, equal to or greater than the trout's value of *trout-superind-rep*). (The competition for hiding places and updating of *cell-available-hiding-places* is described in Sect. 9.11.) Sect. 22.2 provides guidance on estimating values of *cell- available-hiding-places*.

The total number of hiding places in a cell is represented by the cell variable *cell-num-hiding-places*. Some of our knowledge of how trout compete for hiding cover can be incorporated in inSTREAM 7 by considering it while evaluating *cell-num-hiding-places* in the field. This variable represents the number of places in a cell where one adult trout could hide in isolation from other trout. Its value of course depends on the size of trout under consideration. Small juveniles can often use many more hiding places, e.g., crevices in relatively small cobble, so their use of hiding cover is not represented explicitly but implicitly in the trout length survival increase function (Sect. 9.18.1). Hiding places for larger trout are typically scarcer, more quantifiable, and more important to the population. We suggest considering one-year-old and larger trout in evaluating hiding places. Quantitative techniques have been developed for counting hiding places (e.g., Finstad et al. 2007), but *cell-num-hiding-places* can also be evaluated by mapping the estimated density of hiding places in the field.

The survival increase function for hiding is 0.0 for fish engaged in feeding (i.e., *trout-activity* = "drift" or "search"), or for fish using the hiding activity in the absence of available hiding places. For individuals using a hiding place, the survival increase function has the value of the trout parameter ***mort-terr-pred-hiding-factor***. The value of this parameter should depend on what predators are believed most important and their ability to extract fish from cover. For example, where otters are common, ***mort-terr-pred-hiding-factor*** will have a value sufficiently less than 1.0 to represent their ability to extract trout from hiding cover.

9.18.7 Parameter values and submodel exploration

We recommend reviewing the terrestrial predation parameter values (Table 9) for every application of inSTREAM 7. To understand this complex submodel, we present plots showing its response to two key variables at a time. In the following figures, the standard parameter values (for large rivers) are used and trout are not hiding. Variables not explicitly varied in the plots have these standard values: *turbidity* = 5 NTU, *sunlight-irradiance* = 200 W/m^2, ***reach-shading*** = 0.9, *trout-length* = 10 cm, *trout-activity* = "drift" or "search", *cell-velocity* = 20 cm/s, and *cell-escape-dist* = 200 cm. The value of *cell-light* is in all cases calculated from *sunlight-irradiance*, ***reach-shading***, *turbidity*, and *cell-depth* (Sect. 9.10).

An exploration of how terrestrial predation survival varies with depth, fish length, and light (Figure 19) illustrates how survival is relatively high for small trout but depends sharply on depth for trout above ~5 cm length. Decreasing light levels, from day to twilight to night, increase the survival probability but do not alter the shape of the relation between depth and survival.

For a feeding adult trout, depth and distance to escape cover have effects similar in magnitude (Figure 20). Greater depth and lower distance to escape cover both sharply reduce predation risk.

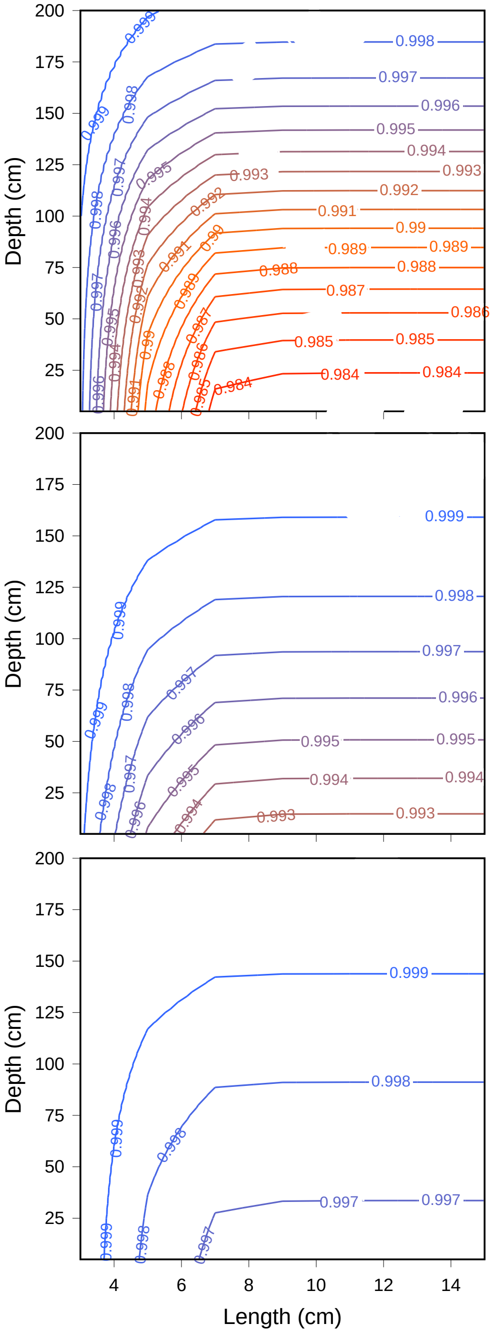


Figure 19: Variation in terrestrial predation survival probability with trout length and cell depth, under three light conditions: top left: *sunlight-irradiance* = 200, characteristic of daytime; top right: *sunlight-irradiance* = 20, characteristic of twilight; and bottom: *sunlight-irradiance* = 0.9, night.

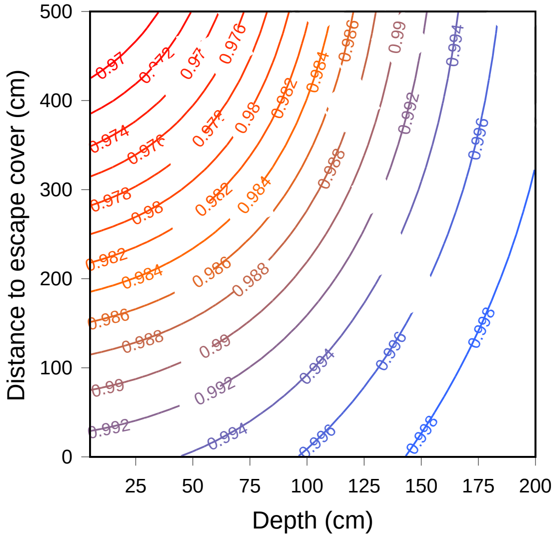


Figure 20: Variation in terrestrial predation survival with depth and distance to escape cover. Trout length is 10 cm, velocity is 20 cm/s, turbidity is 20 NTU, and surface irradiance is 200 W/m^2 (daytime).

Table 9: Parameters and values for terrestrial predation

Parameter name	Definition	Basis	Standard value
reach-terr-pred-min	Daily survival probability for terrestrial predation with the least-protective habitat and activity	Typically estimated via calibration; typical values are 0.90 to 0.99	0.95
mort-terr-pred-L1	The trout length (cm) at which survival of terrestrial predation increases by 0.1	Literature discussed above	6.0
mort-terr-pred-L9	The length at which survival of terrestrial predation increases by 0.9		3.0
mort-terr-pred-D1	The depth (cm) at which survival of terrestrial predation is increased by 0.1	Literature discussed above for *Oncorhynchus* spp.	Small streams: 0.0 Large rivers: 0.0
mort-terr-pred-D9	The depth (cm) at which survival of terrestrial predation is increased by 0.9		Small streams: 140 Large rivers: 200

Parameter name	Definition	Basis	Standard value
mort-terr-pred-V1	The cell velocity (cm/s) at which terrestrial predation survival is increased by 0.1	Judgment	Small streams: 20 Large rivers: 20
mort-terr-pred-V9	The velocity at which terrestrial predation survival is increased by 0.9		Small streams: 100 Large rivers: 300
mort-terr-pred-I1	The value of *cell-light* (W/m^2) at which terrestrial predation survival is increased by 0.1		50
mort-terr-pred-I9	The value of *cell-light* increasing terrestrial predation survival by 0.9		-10
mort-terr-pred-H1	The value of *cell-escape-dist* (cm) at which terrestrial predation survival is increased by 0.1	Judgment; Harvey and White (2017)	200
mort-terr-pred-H9	The value of *cell-escape-dist* increasing terrestrial predation survival by 0.9		-50
mort-terr-pred-hiding-factor	The fraction by which survival is increased by the use of hiding cover.	Judgment, assuming some predators (e.g., otters) can extract trout from hiding cover.	0.8

9.19 Fish predation trout mortality

The fish predation formulation represents mortality due to predation on trout by all piscivorous fish[43]. Many trout populations do not co-occur with other piscivorous fish species, in which case fish predation is cannibalism by large trout. Other populations—often ones affected by large-scale threats such as climate change and invasive species—co-exist with other piscivores. The formulation represents the effect of adult trout density on predation survival, making this survival probability the only component of inSTREAM with direct density dependence. However, parameter values can be chosen to also reflect risk from non-trout fish that are not otherwise represented in inSTREAM.

Especially at sites where trout rarely get larger than 20-30 cm, cannibalism by trout can be relatively rare; e.g., Railsback and Harvey (2001) reported that fewer than 1 percent of adult Cutthroat Trout sampled in Little Jones Creek had juveniles in their guts. However, the risk of

[43] Fish predation survival is in the procedure `survival-fish-pred-for`.

piscivory appears to be an important factor driving habitat selection (e.g., Brown and Moyle 1991): avoiding predation is likely a key reason why small fish prefer shallow water. Also, there have been anecdotal reports of very high cannibalism rates during fry emergence in some salmonids.

Considering the characteristics of piscivorous fish leads to these generalizations about fish predation:

Small trout are generally more vulnerable, and risk may be negligible for large trout. However, the relation between trout size and risk depends on the size and species of predators.

Fish metabolic demands depend on temperature, so their consumption of other fish may be lower at low temperatures.

Piscivorous fish are large and hence at risk from terrestrial predators, so habitat with high risk of terrestrial predation (especially, low depth) can have low risk of fish predation.

Because both predators and prey share the same light conditions, low light may offer less protection than it does for terrestrial predation.

The fish predation survival formulation, like the terrestrial predation one, uses a minimum survival probability representing the riskiest conditions and survival increase functions that represent how fish and habitat characteristics can increase survival probability. The formulation is:

$$S_F = reachFishPredMin + \left((1.0 - reachFishPredMin) \times \left(1.0 - \prod_{i=1}^{i=n} (1.0 - I_i) \right) \right)$$

where S_F is the daily probability of surviving fish predation, ***reach-fish-pred-min*** is a reach parameter representing survival probability in the least-protective combination of habitat and activity, and I_i represents the *n* survival increase functions described below. The value of ***reach-fish-pred-min*** can be calibrated to observations if data representing survival of small trout are available. ***reach-fish-pred-min*** is a reach parameter instead of a trout parameter so it can be varied among reaches with different levels of predation, e.g., those with and without piscivorous fish other than salmonids.

The fish predation submodel includes the following survival increase functions. Suggested parameter values are provided at the end of the section (Table 10). As with the terrestrial predation submodel, new survival increase functions are easily added and existing functions can be turned off by setting their parameter values to yield function values near zero.

There is no survival increase function for distance to escape cover in the fish predation formulation. This decision was made because only small trout are usually vulnerable to aquatic predators, and small trout are capable of hiding in many places that do not offer refuge to adult trout (e.g., between relatively small cobbles) and hence are not well represented by the cell variable for distance to escape cover. There is also no survival increase function for velocity because we assume neither low nor high velocity would give a small trout a consistent advantage over a fish predator.

9.19.1 Trout length

As fish grow, their swimming abilities improve and fewer piscivorous fish are big enough to swallow them. The length survival increase function is therefore an increasing logistic function, the parameters for which depend on the size of the piscivorous fish. Keeley and Grant (2001) provide an empirical relation between the size of piscivorous stream trout and the size of their fish prey. Figure 21 illustrates parameters for sites where fish predation is mainly due to trout 25-30 cm in length.

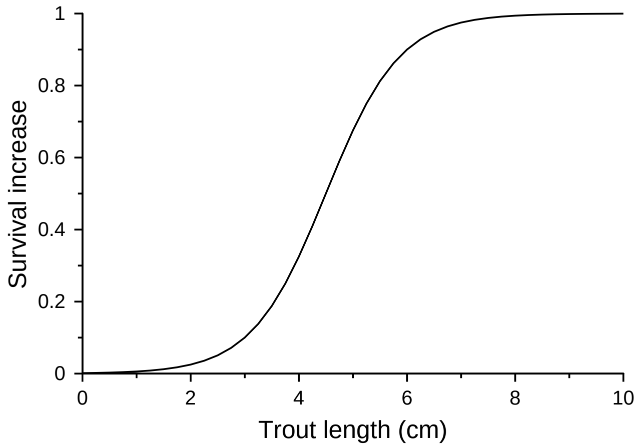


Figure 21: Fish predation survival increase function for trout length, using the parameter values from Table 10.

9.19.2 Depth

Fish predation survival is assumed to be high in water shallow enough to physically exclude large fish or place them at high risk of terrestrial predation. The depth survival increase function (Figure 22) is therefore a decreasing logistic function, with very high survival at depths less than 5 cm and high survival at depths that large trout have been observed to avoid (discussed in Sect. 9.18.2).

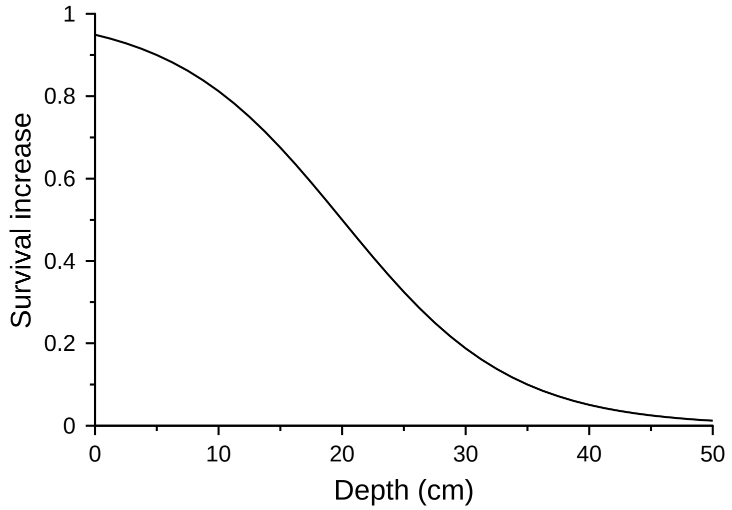


Figure 22: Fish predation survival increase function for depth, using the parameter values from Table 10.

9.19.3 Light

The survival increase function for light represents how cell light irradiance, as determined by surface irradiance, depth, and turbidity, affects fish predation. The relation between light irradiance and piscivory risk is complex but can be informed by the literature. Turbidity and low light levels appear to reduce the ability of piscivorous fish to detect prey fish and thus the encounter rate between predator and prey (Gregory and Levings 1998, Vogel and Beauchamp 1999, DeRobertis et al. 2003). However, the same conditions make potential prey fish less able to detect and escape from predators. Another mechanism that can offset this reduced encounter rate is that turbidity also reduces the vulnerability of piscivorous fish to terrestrial predation, making them more likely to forage in shallow habitat commonly occupied by small fish (Vogel and Beauchamp 1999).

Several publications have observed and reviewed irradiance and turbidity effects on feeding by piscivorous fish. Utne-Palm (2002) reviews literature and theory on how turbidity affects feeding in fish. Hansen et al. (2013) measure reactive distances in piscivorous Chinook Salmon and Cutthroat Trout preying on trout, over a range of surface irradiance in clear water. They observed little effect of irradiance at high levels (> 18-25 lux) and little effect at low irradiance (< 2 lux), but reactive distance dropped by about 50% as irradiance declined from the high to low range. (See Table 13 to relate lux to the irradiance measure used by inSTREAM.) Hansen et al. (2013) also observed this reactive distance over a range of turbidities from 0 to 7 NTU at a surface irradiance of 50 lux. Over this range, reactive distance decreased by about 70%. Gregory and Levings (1998) compared piscivory by fish in adjacent clear and turbid rivers and found piscivory much lower, but still present, in the turbid river. Michel et al. (2020) observed that the rate of predation by warmwater fish on tethered juvenile salmonids increased with temperature, bottom roughness (perhaps reflecting availability of cover for predators), and proximity to predators (which will vary with the density of predators). They also observed higher predation rates in the hour after sunset than during the day.

In consideration of this literature, we suggest parameters for this function that provide no protection from aquatic predation at low daytime irradiances, and a 50 percent reduction in risk at night values (Figure 23). These parameters do not reproduce a higher risk during crepuscular periods as indicated by Michel et al. (2020).

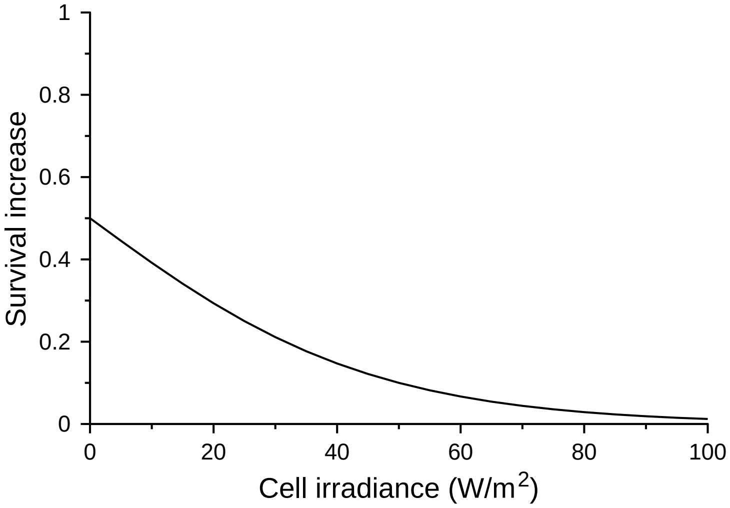


Figure 23: Fish predation survival increase function for cell irradiance, as affected by surface light, turbidity, and depth.

9.19.4 Hiding activity

This function represents the reduction in fish predation risk for trout that select the hiding activity instead of feeding. Unlike the hiding activity function for terrestrial predation (Sect. 9.18.6), this function assumes hiding can reduce risk even for trout not occupying a hiding place. We make this assumption because fish predation is mainly a risk to small trout, and small trout can use widespread hiding cover too small to meet our definition of hiding places (i.e., places where adult trout can hide; Sect. 9.18.6).

The fish predation survival increase function for hiding is 0.0 for fish not using the hiding activity (i.e., *trout-activity* = "drift" or "search") and the value of the parameter ***mort-fish-pred-hiding-factor*** for hiding individuals (*trout-activity* = "hide"). The value of ***mort-fish-pred-hiding-factor*** will depend on what kind of piscivorous fish are present and their ability to explore hiding places. The parameter value can also consider the abundance and quality of hiding cover useable by small fish. Sites with abundant cobble in areas often used by small trout will have a higher value of ***mort-fish-pred-hiding-factor*** than sites where substrate generally lacks hiding cover (e.g., sand, bedrock).

9.19.5 Trout predator density

This function adjusts the fish predation survival probability for the density of trout large enough to be predators[44]. It is, therefore, a mechanism of direct density dependence: higher adult trout abundance reduces the probability of juvenile trout surviving to adulthood. The importance and shape of this function of course depend on whether trout are the only potential fish predators; it represents only predation by simulated trout on other simulated trout, not the effect of other piscivorous fish that may be at a site but not represented in inSTREAM except via the fish

[44] Piscivorous trout density is updated in the procedure `update-logistics`.

predation submodel. If a site has abundant non-salmonid fish predators, then this function can be parameterized so that the survival probability increases little with decreasing adult abundance.

This function assumes that survival probability decreases as the density of potentially piscivorous trout increases. (To avoid excess complexity, we neglect any effect of prey fish density on per-trout risk.) "Density of potentially piscivorous trout" is defined at the reach scale (reach variable *pisciv-trout-density*). It is the number of trout in the reach that have *trout-length* greater than both (a) the length of the trout calculating survival and (b) the fish parameter ***trout-pisciv-length***, divided by the total wetted area of the reach (the sum of *cell-area* over all the reach's cells with *cell-depth* > 0, cm^2). ***Trout-pisciv-length*** is defined as the minimum length (cm) of trout assumed able to eat other trout; its value can vary among trout species to reflect differences in piscivory. The density of potentially piscivorous trout is updated at the start of each time step; changes within a time step due to mortality, movement among reaches, or growth of large trout are neglected. If a reach is completely dry (none of its cells have depth > 0.0), then *pisciv-trout-density* is set to zero.

The piscivorous trout density function can be parameterized by thinking about the number of trout within a specific area that would represent high and low predator densities. The standard values in Table 10, used in Figure 24, were estimated by assuming the survival increase would be low (0.1) with 5 piscivorous trout per 10×10 m area and high (0.9) with only 0.5 trout per the same area.

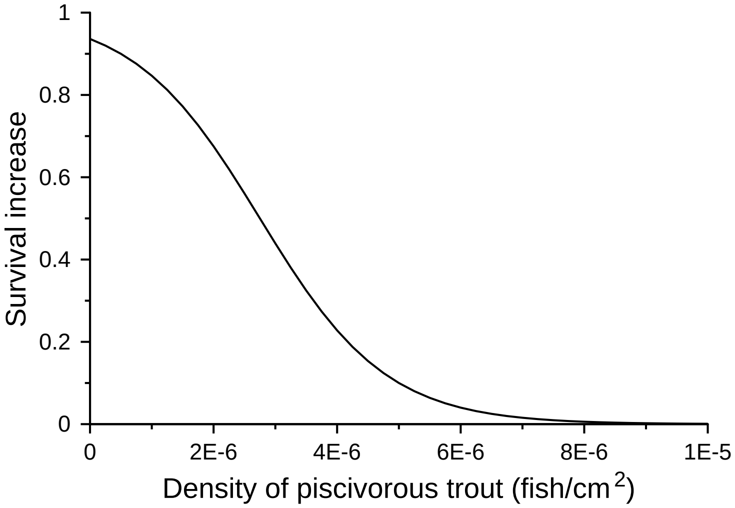


Figure 24: Fish predation survival increase function for piscivorous trout density.

9.19.6 Temperature

This survival increase function assumes that survival of fish predation is higher at low temperatures. This assumption is based on two mechanisms. When other trout are the only predators, the function represents how low temperatures reduce the metabolic demands and, therefore, feeding activity of piscivorous fish. In this case, the function can be based on the bioenergetics of the trout predators, using a function (Figure 25) that approximates the decline in maximum food consumption with declining temperature assumed in the feeding submodels (Sect. 9.24). Our standard parameters (Table 10, Figure 25) are based on this mechanism.

However, the second mechanism is important when fish predation is dominated by non-trout piscivores that do not function as well as trout at low temperatures. Parameter values can be chosen to reflect how metabolic rates and swimming performance of less cold-adapted predators drop at low temperatures.

The temperature function is a decreasing logistic function of the reach variable *temperature*.

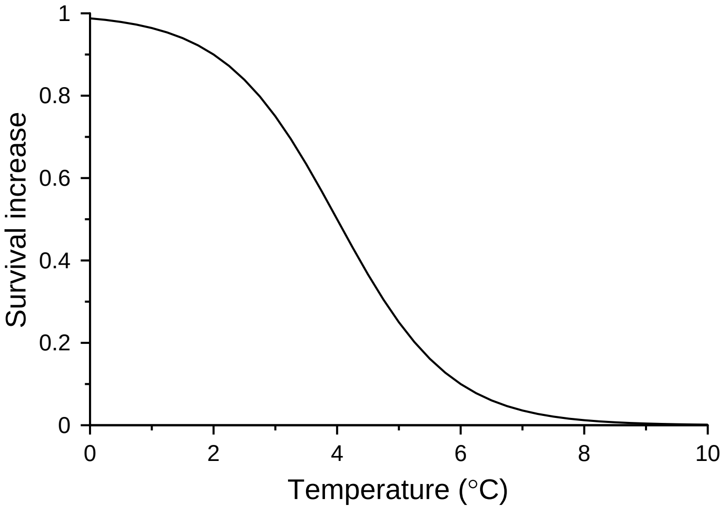


Figure 25: Fish predation survival increase function for reach temperature, parameterized assuming trout are the only predators.

9.19.7 Parameter values and submodel exploration

Here we summarize the standard values for fish predation parameters (in Table 10) and explore the full fish predation submodel. The parameter values deserve reconsideration for each application of inSTREAM, especially when factors such as the presence of non-salmonid piscivorous fish affect the mechanisms represented in the submodel.

Especially to understand habitat and activity selection, it is valuable to not only understand this submodel but also to compare it directly to the terrestrial predation submodel (Sect. 9.18). Terrestrial and fish predation are generally two of the most important factors driving habitat and activity selection, and, for some fish, act in opposite ways: habitat safe from one kind of predation puts individuals at risk of the other. To make this comparison, we plot fish predation survival probability for feeding trout under the same conditions and over the same variable ranges for which terrestrial predation was explored in Sect. 9.18.7.

The plots of how fish predation survival varies with depth, fish length, and light (Figure 26) show key differences from terrestrial predation (Figure 19). Survival of fish predation is dominated by trout length, not cell depth: fish predation is important only for small trout.

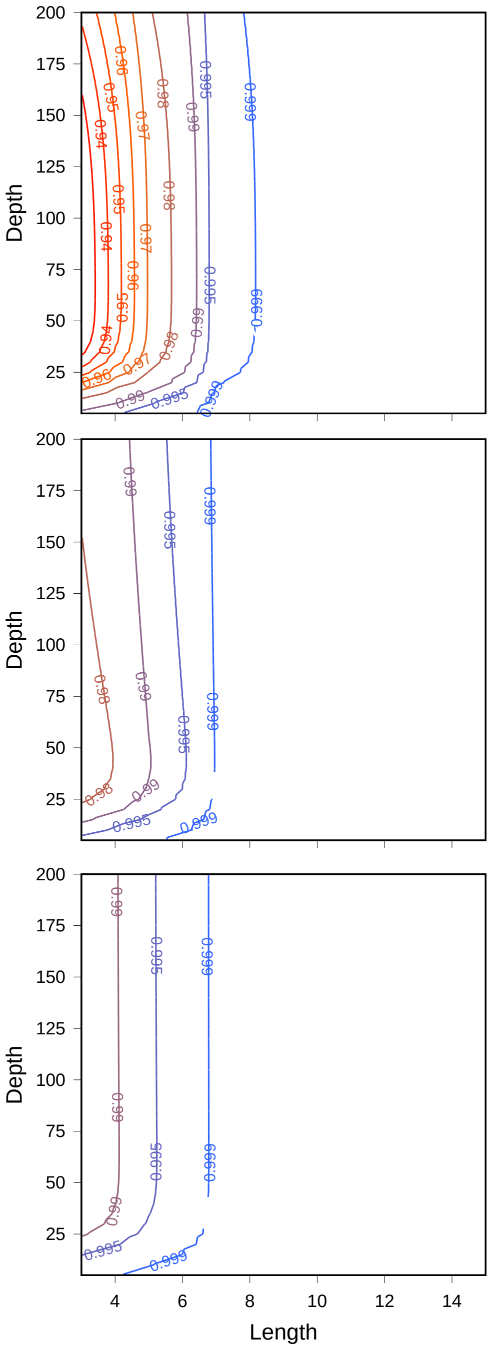


Figure 26: Variation in fish predation survival probability with trout length and cell depth, under three light conditions: top left: *sunlight-irradiance* = 200, characteristic of daytime; top right: *sunlight-irradiance* = 20, characteristic of twilight; and bottom: *sunlight-irradiance* = 0.9, night.

Table 10: Parameters and values for fish predation

Parameter name	Definition	Basis	Standard value
reach-fish-pred-min	Daily survival probability for fish predation with the least-protective habitat and activity	Typically estimated via calibration; typical values are 0.9 to 0.95	0.9
mort-fish-pred-L1	The trout length (cm) at which survival of fish predation increases by 0.1	Literature discussed above	3.0
mort-fish-pred-L9	The length at which survival of fish predation increases by 0.9		6.0
mort-fish-pred-D1	The depth (cm) at which survival of fish predation is increased by 0.1	Judgment based on large fish use of depth	35
mort-fish-pred-D9	The depth (cm) at which survival of fish predation is increased by 0.9		5.0
mort-fish-pred-I1	The value of *cell-light* (W/m2) at which fish predation survival is increased by 0.1		50
mort-fish-pred-I9	The value of *cell-light* increasing fish predation survival by 0.9		-50
mort-fish-pred-hiding-factor	The fraction by which survival is increased by use of hiding activity	Judgment	If hiding cover for small trout is abundant: 0.7 If good hiding cover is scarce: 0.5
trout-pisciv-length	The length (cm) above which a trout is assumed capable of eating other trout		15
mort-fish-pred-P1	The density (fish/cm^2) of piscivorous trout at which survival is increased by 0.1	Judgment	5.0E-6
mort-fish-pred-P9	The density of piscivorous trout at which survival is increased by 0.9		5.0E-7

Parameter name	Definition	Basis	Standard value
mort-fish-pred-T1	The reach temperature (°C) at which survival is increased by 0.1	Relation between maximum food consumption and temperature for trout	6.0
mort-fish-pred-T9	The temperature at which survival is increased by 0.9		2.0

9.20 Growth

The growth submodel updates a trout's weight, length, and condition variables, and possibly its superindividual status, to reflect the growth obtained from the habitat cell and activity chosen on the current time step[45]. It uses the daily growth rate (g/d) calculated in the growth rate submodel (Sect. 9.21). The variable *trout-growth* is the grams of growth during the time step, calculated simply as the daily growth rate times *step-length*.

In the first step of growth, weight is updated by adding *trout-growth* to *trout-weight*; if the result is negative then *trout-weight* is set to zero.

Updating length and condition are more complex, even though inSTREAM highly simplifies the relation between a fish's weight and its length. Abundant literature indicates that the allocation of energy intake between growth in body structure and energy reserves is an important adaptive behavior, but useful models of this complex behavior are not yet available (discussed further below). Instead, inSTREAM updates length using the simple assumption of Van Winkle et al. (1996) that growth is allocated only to weight if a fish's weight is less than a "healthy" weight for its length, and otherwise is allocated to growth in both length and weight.

The method for updating length adopted from Van Winkle et al. (1996) also uses their nonstandard definition of a condition factor. In fisheries science, a condition factor is an index of a fish's weight relative to its length. A high condition factor indicates high weight relative to length, and therefore less vulnerability to starvation or disease during periods of negative growth. The condition factor variable used in inSTREAM (*trout-condition*) can be considered the fraction of "healthy" weight a fish is, given its length, and is never greater than 1.0.

"Healthy" weight (*trout-healthy-weight*, g) is calculated from length (*trout-length*, the length before being updated) using a standard allometric relation:

$$troutHealthyWeight = troutWeightA \times troutLength^{troutWeightB}$$

where ***trout-weight-A*** and ***trout-weight-B*** are trout parameters. These parameters can be estimated from field data, e.g., by identifying the individuals with relatively high weight per length (e.g., the individuals with greater than median values of weight divided by length cubed). Simulated growth is quite sensitive to the values of these parameters, so changing them affects model calibration and results. We provide example values in Table 11.

[45] The growth submodel is programmed in the procedure `grow`.

If the updated value of *trout-weight* is greater than *trout-healthy-weight*, then the trout grows in length. *Trout-condition* is set to 1.0 and length is set to the length of a healthy fish with the updated weight:

$$troutLength = \left(\frac{troutWeight}{troutWeightA}\right)^{(troutWeightB^{-1})}$$

If instead the updated value of *trout-weight* is less than or equal to *trout-healthy-weight*, then the trout does not grow in length. The value of *trout-length* is unchanged and *trout-condition* is set to *trout-weight* divided by *trout-healthy-weight*.

This simple formulation succeeds in producing reasonably realistic patterns of trout growth under many conditions. However, it has several noteworthy limitations:

Fish cannot preserve a high-energy-reserve condition. Fish will have a condition of 1.0 only on time steps with positive growth. Even if a fish has eaten well for many days in succession, its value of *trout-condition* can only be as high as 1.0. Further, any time step on which growth is negative (e.g., when the fish hide) causes condition to fall below 1.0. Trout in inSTREAM 7 can hide often (Sect. 9.13) and hence often have condition below 1.0.

This weight-based condition factor is not the best predictor of starvation mortality (Sect. 9.17).

This formulation locks in a length-weight relationship for growing fish. Therefore, inSTREAM will automatically reproduce the length-weight relationship defined by ***trout-weight-A*** and ***trout-weight-B*** during periods of growth, but is not capable of reproducing other patterns (e.g., seasonal variation or variation among individuals or sites) in length-weight relations.

The energetics of reproduction are not considered. While inSTREAM does simulate weight loss due to spawning (Sect. 9.28), it does not model storage of energy for gonad development and how gonad production affects length and weight and, therefore, behavior.

Our experience with inSTREAM indicates that its simplified method for updating length is sufficient for the model's purpose of predicting population responses and avoids a great amount of complexity. Investigating how to address the above limitations and how they affect inSTREAM's results remains a research interest.

The final step in the growth submodel determines whether the trout is a superindividual ready to be split into regular individuals. If the new value of *trout-length* is greater than the trout parameter ***trout-superind-max-length*** (cm), then it executes the submodel that splits it (Sect. 9.39).

Table 11: Parameters and values for growth

Parameter name	Definition	Basis	Value
trout-energy-density	The ratio (J/g) between net energy intake and change in weight	Literature summarized by Hanson et al. (1997)	5900
trout-weight-A ***trout-weight-B***	The multiplier and exponent in the allometric relation for weight of a "healthy" trout	Cutthroat Trout, Little Jones Creek, Del Norte County,	0.0124 2.98

		California (Railsback and Harvey 2001)	
trout-weight-A ***trout-weight-B***		Rainbow Trout, Tule River, Tulare County, California (Van Winkle et al. 1996)	0.0134 2.96
trout-weight-A ***trout-weight-B***		Brown Trout, Tule River, Tulare County, California (Van Winkle et al. 1996)	0.0123 2.97
trout-weight-A ***trout-weight-B***		Rainbow Trout, Green River below Flaming Gorge Reservoir, Utah (Railsback et al. 2006)	0.0185 2.90
trout-weight-A ***trout-weight-B***		Brown Trout, Green River below Flaming Gorge Reservoir, Utah (Railsback et al. 2006)	0.0157 2.91

9.21 Growth rate

The growth rate submodel calculates the growth rate (rate of change in body weight, g/d, which can be negative) that a trout would experience in a particular cell, using a particular activity, on a specific time step[46]. Growth is modeled as proportional to net energy intake rate (NEI), the difference between the energy intake from feeding and metabolic costs. The submodel works in three steps.

The first step calculates gross energy intake (J) from feeding, which depends on the trout's activity. The hiding activity provides zero gross energy intake. The drift and search feeding activities have their own submodels to calculate gross intake (sects. 9.22 and 9.23). Food intake rate (g/d) from those submodels is converted to gross energy intake rate (J/d) by multiplying food intake rate by the reach parameter ***reach-prey-energy-density***, which represents the energy content of fish prey (J/g).

The second step calculates the respiration cost rate, using the respiration cost submodel (Sect. 9.25). Respiration rate depends in part on swimming speed, which depends on activity as explained in Sect. 9.13.

The third step calculates NEI and converts it to growth rate. NEI is equal to the gross energy intake rate calculated in the first step, minus the respiration costs determined in the second step. This difference (J/d) is converted to growth rate by dividing it by the trout parameter ***trout-energy-density*** (J/g). This parameter therefore represents the growth (g) a trout gets from a unit of net energy intake (J). The literature offers direct measurements of fish energy density (e.g., Rottiers and Tucker 1982; Johnson et al. 2017). The energy content of salmonids can vary among individuals (typically increasing with age) and over time (e.g., seasonally; Sloat and Reeves 2014). While such variation may have adaptive consequences, inSTREAM currently neglects it except for allowing different values of ***trout-energy-density*** for different species or study sites.

[46] The growth rate submodel is programmed in procedure `growth-rate-for`.

Table 12: Parameters and values for growth rate

Parameter name	Definition	Basis	Standard value
reach-prey-energy-density	The energy content of fish prey (both search and drift; J/g)	Literature summarized by Hanson et al. (1997)	2500
trout-energy-density	The ratio (J/g) between net energy intake and change in weight	Literature summarized by Hanson et al. (1997)	5900

9.22 Drift feeding

The drift feeding submodel calculates the food intake rate (g/d) in a particular cell for a trout using the drift feeding strategy[47]. Drift (or "sit-and-wait") feeding is widely assumed the dominant feeding strategy of stream salmonids. The submodel assumes that drift intake rate is limited by the rate at which the trout can capture drift prey, the physiological maximum food intake rate, or the drift prey available in the cell. At its highest level, the submodel is:

$$I_D = min\left(C_D, C_{stepmax}, cellAvailableDrift\right)$$

where I_D is the daily rate of drift food intake (g/d), C_D is the modeled rate of drift food capture, $C_{stepmax}$ is the maximum consumption rate for the time step (Sect. 9.24), and *cell-available-drift* is the drift food availability rate in the cell at the current time, considering the food production rate and depletion by larger fish (Sect. 9.11.2). (If the trout is a superindividual, the value of *cell-available-drift* is divided by the trout's value of *trout-superind-rep* so it represents drift availability per fish.) The rest of this section describes the drift capture rate submodel.

The drift feeding submodel of inSTREAM 7 is unique but builds on an extensive base of data and theory going at least as far back as Fausch (1984). This literature has established a common paradigm of drift intake depending on three processes: the rate at which flow carries prey past a stationary trout; the ability of trout to detect the prey, which is typically modeled as a "reaction distance"; and the ability of trout to capture prey once detected.

The ability of trout to detect and capture prey is widely understood to depend on several factors, many of which interact:

Trout size: larger fish have larger eyes and therefore better detection ability, and better ability to capture prey and return to the feeding station.

Water velocity: higher velocities make it more difficult to capture prey and return to the feeding station.

[47] Drift food intake is programmed in the procedure `drift-intake-for`.

Turbidity: while low levels of turbidity can theoretically make small prey easier to see (by increasing contrast between prey and background; Utne-Palm 2002), higher levels above a threshold sharply reduce detection ability.

Prey size: larger prey are easier to detect.

Light intensity: while trout are able to feed at very low light levels, their detection ability increases rapidly as light levels approach daytime values.

InSTREAM 7 represents all these factors except prey size. Prey size is neglected to avoid the complexity of representing it; instead, the model treats all prey as having a single, unspecified, size.

While an extensive literature addresses the effects of light and turbidity on the ability of fish to detect prey (e.g., Hansen and Beauchamp 2015 and the review of Utne-Palm 2002), little of it is directly applicable to the light function. Reasons for caution in using literature models include empirical studies that used prey more visible than the aquatic macroinvertebrates assumed in inSTREAM (e.g., food items floating on the surface, or fish as prey) and limitations on reaction distance imposed by the experimental apparatus. While some studies separately measured the effects of surface light and turbidity (e.g., Hansen et al. 2013), we found only one study (Vinyard and O'Brien 1976) that varied both in combination.

Another limitation is that much of the experimental literature measures light as illuminance in units of lux; illuminance is a measure of light visible to the human eye and is not directly convertible to the irradiance measure used in inSTREAM. To make use of such literature, we characterize a range of light conditions using both illuminance and irradiance (Table 13). The illuminance values are from the Wikipedia article on lux (https://en.wikipedia.org/wiki/Lux, accessed 3 May 2019), and the irradiance values are from our light submodel (Sect. 9.9).

Table 13: Illuminance and irradiance values typifying a range of light conditions.

Condition	Typical illuminance values (lux)	Typical irradiance values (W/m^2)
Night without bright moon	0.002	0.9
Sunrise or sunset	400	20-25
Full daylight	10-25,000	200-600

Despite the above limitations, the literature clearly identifies several patterns useful for modeling the effects of turbidity and light:

Reaction distance is unaffected by light intensity above a threshold value. Hansen et al. (2013) found reaction distance of Cutthroat Trout and Chinook Salmon preying on small fish to stop increasing with illuminance at a threshold of ~20 lux. Vinyard and O'Brien (1976) found the reaction distance of juvenile bluegill hunting daphnia to stop increasing at a threshold of about 10 lux.

Low levels of turbidity appear not to significantly reduce reaction distance; Hansen et al. (2013) found a threshold of 2 NTU for turbidity effects with fish prey, while previous versions of inSTREAM used a threshold of 5 NTU that was based on the results of Sweka and Hartman (2001).

As light intensity decreases below the threshold at which it has no effect, reaction distance decreases sharply but then again plateaus at a lower threshold. Hansen et al. (2013) found reaction distance to decrease sharply by half as illuminance decreased from ~20 to 5 lux, but to decrease little as illuminance further decreased.

The effect of increasing turbidity also is strong just above the no-effect threshold but declines as turbidity increases. The results of Hansen et al. (2013) and Sweka and Hartman (2001; Figure 28) show a steep decline in reaction distance at turbidity levels above the no-effect threshold but reaction distance then appears to reach a lower threshold at high turbidity levels, perhaps because fish use senses other than vision to find prey.

Turbidity has strong effects on reaction distance at values (5-20 NTU) that do not reduce daytime light intensity to levels that would by themselves affect reaction distance. For example, with the standard parameter values of Sect. 9.10 a cell 50 cm deep, with *sunlight-irradiance* = 200 and *turbidity* = 20 NTU, has a *cell-light* value of 68 W/m^2, well above the twilight levels where reaction distance starts to decline because of light intensity.

These patterns lead us to conclude, as did Vinyard and O'Brien (1976), that irradiance and turbidity must be treated as having separate and independent effects on reaction distance even though irradiance is in part a function of turbidity.

The conceptual model of drift feeding is that a trout remains stationary (except when capturing prey) and detects the prey carried by the flow through a "capture area" perpendicular to the flow, and successfully captures some fraction of the detected prey. This capture area depends on the reaction distance and, sometimes, water depth. We treat the capture area as a rectangle with width of two times the reaction distance (the fish can detect prey up to the reaction distance in either direction); its height is the lower of reaction distance and depth. The drift intake model is:

$$C_D = \left(2 \times D_r \times min(D_r, cellDepth)\right) \times driftConcentration \times cellVelocity \times 86400 \\ \times captureSuccess$$

where C_D is the intake (capture) rate of drift prey (g/d), D_r is the reaction distance (cm), ***reach-drift-conc*** is a reach parameter representing the prey concentration (g/cm^3; Sect. 9.11.2), 86400 converts units from per second to per day, and *capture-success* is the fraction of detected prey successfully captured.

The parameters of the drift feeding submodel have strong effects on the range of velocities over which simulated trout can obtain positive growth and, therefore, on the relations predicted by inSTREAM between flow and trout populations (Sect. 26.2). We recommend careful consideration and testing before any modification of parameter values. Further, the maximum sustainable swimming speed submodel (Sect. 9.26) strongly affects drift feeding, so any changes to that submodel should carefully consider effects on drift feeding.

9.22.1 Reaction distance

The reaction distance D_r is modeled as:

$$D_r = troutLengthTerm \times turbidityFunction \times lightFunction.$$

The trout length term calculates reaction distance when unaffected by turbidity and light. It is based on empirical studies by Schmidt and O'Brien (1982; see also O'Brien et al. 2004) who measured how detection distance in a stream salmonid (Arctic Grayling) varied with fish and prey size. These experiments used zooplankton as prey whereas we assume trout represented by inSTREAM are more often consuming larger invertebrates, but the zooplankton were large and the results have been used successfully as the basis of drift feeding models by Hughes (1992) and Hughes et al. (2003). Schmidt and O'Brien (1982) measured detection distance of fish with lengths from 3 to 13 cm, during daylight and night conditions, and for a variety of zooplankton prey sizes. Only daylight observations for 0.2 cm prey (the largest) are used here.

These observations can be represented with a linear model having a slope of 2.0 and intercept of 4.0 cm (Figure 27). This linear model is not a regression fit to the data of Schmidt and O'Brien (1982), and in fact a logarithmic equation fits the data more closely. The linear model shown in Figure 27 was chosen for several reasons. First, it captures the fact that very small trout cannot use as wide a range of prey sizes as larger trout can, a process not otherwise represented in the feeding model. Second, a logarithmic fit to these data predicts negative detection distances for trout lengths less than 2 cm and does not reproduce the observations of Hughes et al. (2003) that detection distance continues to increase to over 100 cm for very large trout. Finally, pre-calibration of the growth model was used to select the intercept and slope of the linear model (parameters ***trout-react-dist-A-A*** and ***trout-react-dist-B***, defined below). The pre-calibration analysis indicated that the growth rates of very small trout are very sensitive to the intercept. An intercept of 4.0 was found to provide realistic growth of very small trout at drift food availability values that also produce realistic growth in larger trout.

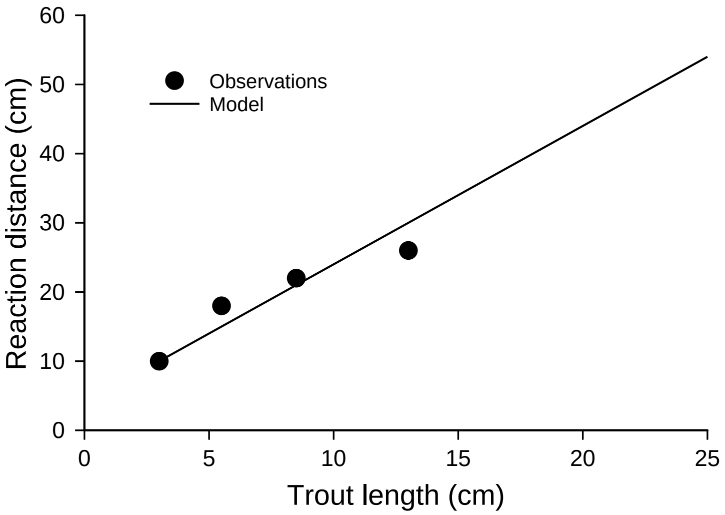


Figure 27: Reaction distance as a function of trout length: observations of Schmidt and O'Brien (1982) and the model with parameters in Table 14.

The trout length function therefore is:

$$troutLengthFunction = troutReactDistA + troutReactDistB \times troutLength$$

where ***trout-react-dist-A*** and ***trout-react-dist-B*** are trout parameters with values in Table 14.

The turbidity function is based on several empirical studies. The primary effect of turbidity on drift feeding appears to be reducing the ability of fish to detect prey: Sweka and Hartman (2001) observed that as turbidity increased the frequency of prey detection by trout decreased but the frequency of attacking and capturing detected prey did not. Sweka and Hartman (2001) conducted a laboratory experiment that appears to be a fairly clear test of the effects of turbidity on the ability of fish to detect prey, over a range of 3-40 NTUs. They developed a curve for how detection distance decreases with turbidity, for 14 cm brook trout feeding on large (1.0 cm) floating prey. The function used by inSTREAM for relative detection distance (the fractional reduction in detection distance due to turbidity; Figure 28) is based on the data of Sweka and Hartman (2001) but differs from their curve in two ways.

First, inSTREAM assumes that turbidity has no effect at values below a threshold of 5 NTUs (defined by the parameter ***trout-turbid-threshold***, NTU). The curve of Sweka and Hartman (2001) has a steep gradient at low turbidity levels, which would make feeding success very sensitive to low turbidity values. However, none of the literature cited above shows a clear negative effect of turbidity at levels below 5 NTUs. Another reason for assuming a turbidity threshold is to avoid making inSTREAM highly sensitive to low turbidity levels, which are hard to measure or estimate accurately.

The second change is the addition of a minimum detection distance. The data of Sweka and Hartman (2001) indicate that detection distance does not go completely to zero as turbidity reaches levels well above 50 NTUs (Figure 28). Our own laboratory studies (Harvey and White 2008) indicate that trout continue to capture some drift, especially near the stream bed, at turbidity levels exceeding 150 NTUs. Therefore, inSTREAM includes a parameter ***trout-turbid-min*** (unitless) which limits the effect of turbidity on detection distance. (This formulation neglects the cessation of feeding at extremely high turbidity values—400 NTU—observed by Harvey and White 2008.)

The turbidity function in the equation for reaction distance assumes that if *turbidity* <= ***trout-turbid-threshold*** then *turbidity-function* is 1.0; otherwise:

$$
\begin{aligned}
turbidityFunction \\
&= troutTurbidMin \\
&+ (1.0 - troutTurbidMin) \\
&\times \exp\big(troutTurbidExp \times (turbidity - troutTurbidThreshold)\big).
\end{aligned}
$$

The trout parameter ***trout-turbid-exp*** has a standard value of -0.0116 fit to the data of Sweka and Hartman (2001; Figure 28). This equation causes the function to asymptotically approach ***trout-turbid-min*** as turbidity increases.

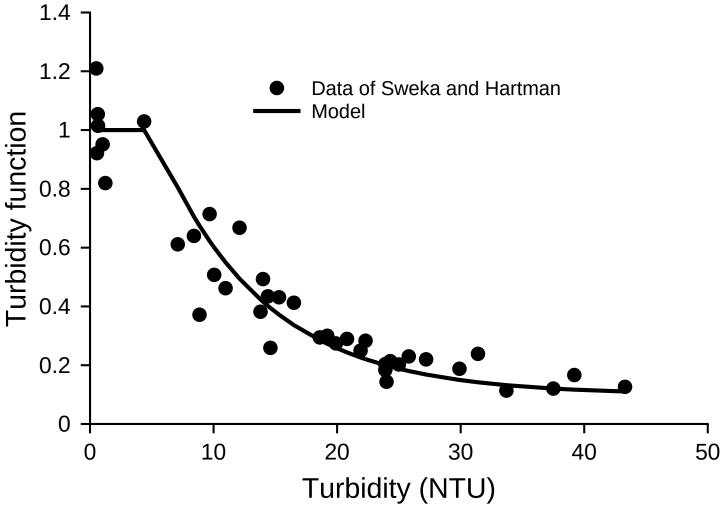


Figure 28: Response of reaction distance (as a fraction of the zero-turbidity values) to turbidity. The symbols are observations by Sweka and Hartman (2001) and the curve is the turbidity function.

The light function is similar to the turbidity function except that reaction distance increases as irradiance increases. The patterns listed above suggest a function in which reaction distance (1) is not affected by light intensity above an upper threshold that corresponds to twilight conditions, (2) decreases rapidly as light decreases below the upper threshold, but (3) reaches a plateau as light decreases to night levels. The study by Vinyard and O'Brien (1976) of bluegill reacting to *Daphnia* prey found a steep decrease in reaction distance as illuminance decreased from 10 to 0.7 lux, corresponding to twilight conditions. The study by Hansen et al. (2013) of Cutthroat Trout and Chinook Salmon juveniles responding to trout prey found a similar steep decrease in reaction distance as illuminance decreased from ~40 to 8 lux, but little further decrease at illuminance down to 0.01 lux. The light function therefore uses the same approach as the turbidity function. If *cell-light* >= ***trout-light-threshold*** then *light-function* is 1.0; otherwise this equation is used:

$$\begin{aligned} F_L = {} & troutLightMin \\ & + (1.0 - troutLightMin) \\ & \times exp\big(troutLightExp \times (troutLightThreshold - cellLight)\big) \end{aligned}$$

where F_L is the light function in the equation for reaction distance, and ***trout-light-min*** and ***trout-turbid-exp*** are trout parameters defining the minimum function value and the rate of decrease. The parameter values in Table 14 reproduce the patterns identified in this paragraph: they cause the light function to approach 0.5 as light decreases toward zero.

Figure 29 illustrates the combined effects of light and turbidity on reaction distance.

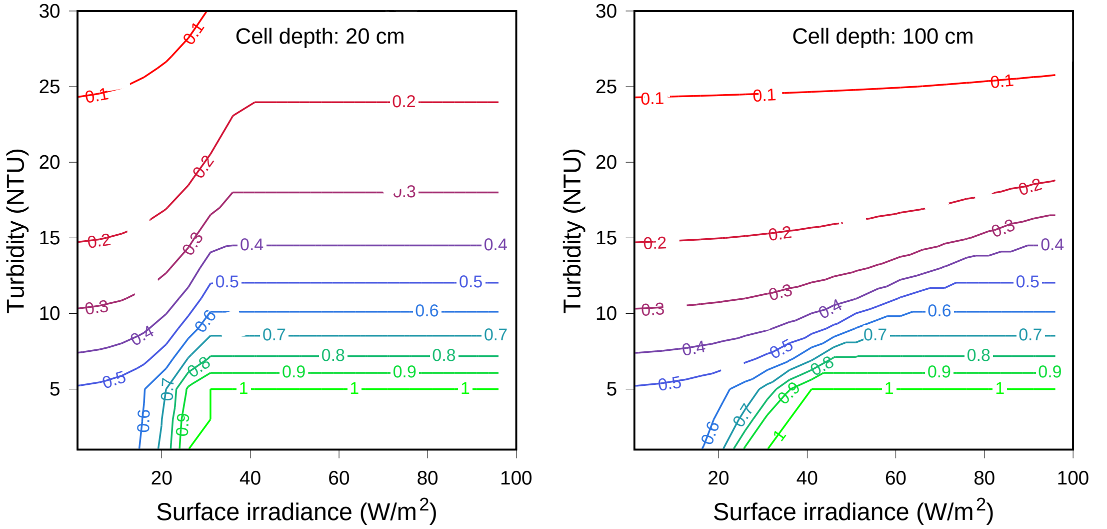


Figure 29: Combined effects of irradiance and turbidity on reaction distance for drift feeding. Contours indicate *turbidity-function* ×*light-function* in cells with depth of 20 cm (left) and 100 cm (right), using standard cell light parameter values. The light function is driven by *cell-light*, which depends on depth and turbidity as well as surface irradiance.

9.22.2 Capture success

While the capture area represents the area over which drift-feeding trout can detect prey, capture success represents the fraction of detected prey actually caught. Capture success is assumed to be a function mainly of water velocity. Fish must be able to intercept the prey, capture it, and return to their feeding station. Higher velocities make maneuvering quickly enough to capture prey more difficult, and swimming longer distances to intercept prey requires more energy (Hughes et al. 2003). Capture success is also affected by low temperatures, which reduce the ability to maneuver and swim rapidly.

Capture success by drift-feeding salmonids has been evaluated in several studies, most conducted in laboratory chambers. Previous versions of inSTREAM based their capture success submodel mainly on the study by Hill and Grossman (1993), who measured capture success for Rainbow Trout feeding on 0.2-cm prey. The trout had lengths of 6 and 10 cm, and measurements were made at 5 and 15ºC with water velocities ranging from 0 to 40 cm/s. Capture success was evaluated as the fraction of prey caught, within the fish's reaction distance. Hill and Grossman (1993) approximated the reaction distance as 2.5 times the fish's standard length, which is fairly close to the reaction distance used in inSTREAM (Figure 30). Hill and Grossman measured capture success within each of three ranges: the inner 20% of the reaction distance, 20-60% of reaction distance, and 60-100% of reaction distance. We evaluate their capture success by averaging these values over the entire reaction distance.

Subsequently, Nislow et al. (1999) evaluated capture success of newly emerged Atlantic Salmon fry via field observations. Additional laboratory observations were made Piccolo et al. (2008; juvenile Coho Salmon and Steelhead), Donofrio et al. (2018; juvenile Chinook Salmon), Bozeman and Grossman (2019; adult Arctic Grayling), and Johansen et al. (2020; adult Rainbow Trout). The only of these studies that directly compared two species (Piccolo et al. 2008) found no substantial difference between Coho Salmon and Steelhead in the relation between capture success and velocity. We chose not to consider the results of Nislow et al. (1999) because they studied newly-emerged fry which appear to have limited swimming ability, and those of Johansen et al. (2020) because they used an especially artificial laboratory setup and (apparently) unexercised hatchery fish.

We examined the capture success results of these studies, scaling velocity by maximum sustainable swimming speed of the fish (Figure 30). Maximum sustainable swim speed (a function of fish length and water temperature; Sect. 9.26.) is useful for modeling capture success because it includes the effects of both fish length and temperature on swimming ability, and we assume swimming ability is the main factor driving capture success. There is some variation among the study results, likely due in part to variation in experimental methods such as laboratory chamber size and configuration, prey size, light conditions, and fish swimming condition. Even so, it is clear that capture success is high and almost unaffected by velocities up to about 50% of maximum sustainable swimming speed, then decreases rapidly with velocity, yet remains well above zero at maximum swimming speed.

Capture success is therefore modeled as a decreasing logistic function of the ratio of cell velocity (*not* the trout's swimming speed, which is often reduced by use of velocity shelter; Sect. 9.13) to maximum sustainable swimming speed. The parameters for the logistic function illustrated in Figure 30 (***trout-capture-R1*** and ***trout-capture-R9***) have values given in Table 14.

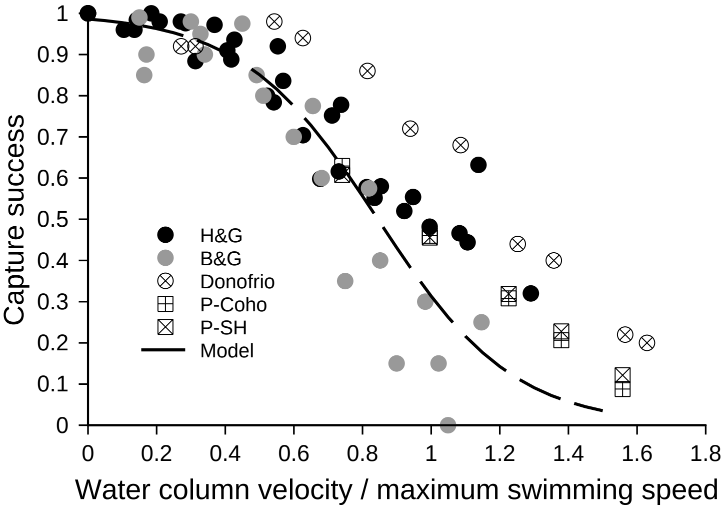


Figure 30: Capture success model and the observations it was based on. H&G: Hill and Grossman (1993), B&G: Bozeman and Grossman (2019), Donofrio: Donofrio et al. (2018), P-Coho: Piccolo et al. (2008) results for Coho Salmon, and P-SH: Piccolo et al. results for Steelhead.

Table 14: Parameters and values for the drift feeding submodel.

Parameter	Meaning and units	Standard value
trout-react-dist-A	Constant in the linear equation for reaction distance as a function of trout length (cm)	4.0
trout-react-dist-B	Multiplier in the equation for reaction distance as a function of trout length (unitless)	2.0
trout-turbid-threshold	Value (NTU) below which turbidity has no effect on reaction distance	5.0
trout-turbid-min	Minimum value of the turbidity function for reaction distance (unitless)	0.1
trout-turbid-exp	Exponential decrease rate for effect of turbidity on reaction distance (NTU^{-1})	-0.116
trout-light-threshold	Value (W/m^2) above which irradiance has no effect on reaction distance	20
trout-light-min	Minimum value of the light function for reaction distance (unitless)	0.5
trout-light-exp	Exponential decrease rate for effect of irradiance on reaction distance (m^2/W)	-0.2
trout-capture-R1	Ratio of *cell-velocity* / trout maximum sustainable swim speed at which capture success is 0.1 (unitless)	1.3
trout-capture-R9	Ratio of *cell-velocity* / maximum sustainable swim speed at which capture success is 0.9 (unitless)	0.4

The drift intake and growth submodels together produce complex responses of growth to many habitat and trout variables. Figure 31 illustrates some of these responses[48].

[48] Figure 31 was produced with output of the procedure `write-growth-report`, which is in the separate file of inSTREAM test code. This procedure reports growth rates for all combinations of selected values for many habitat and trout variables that affect growth.

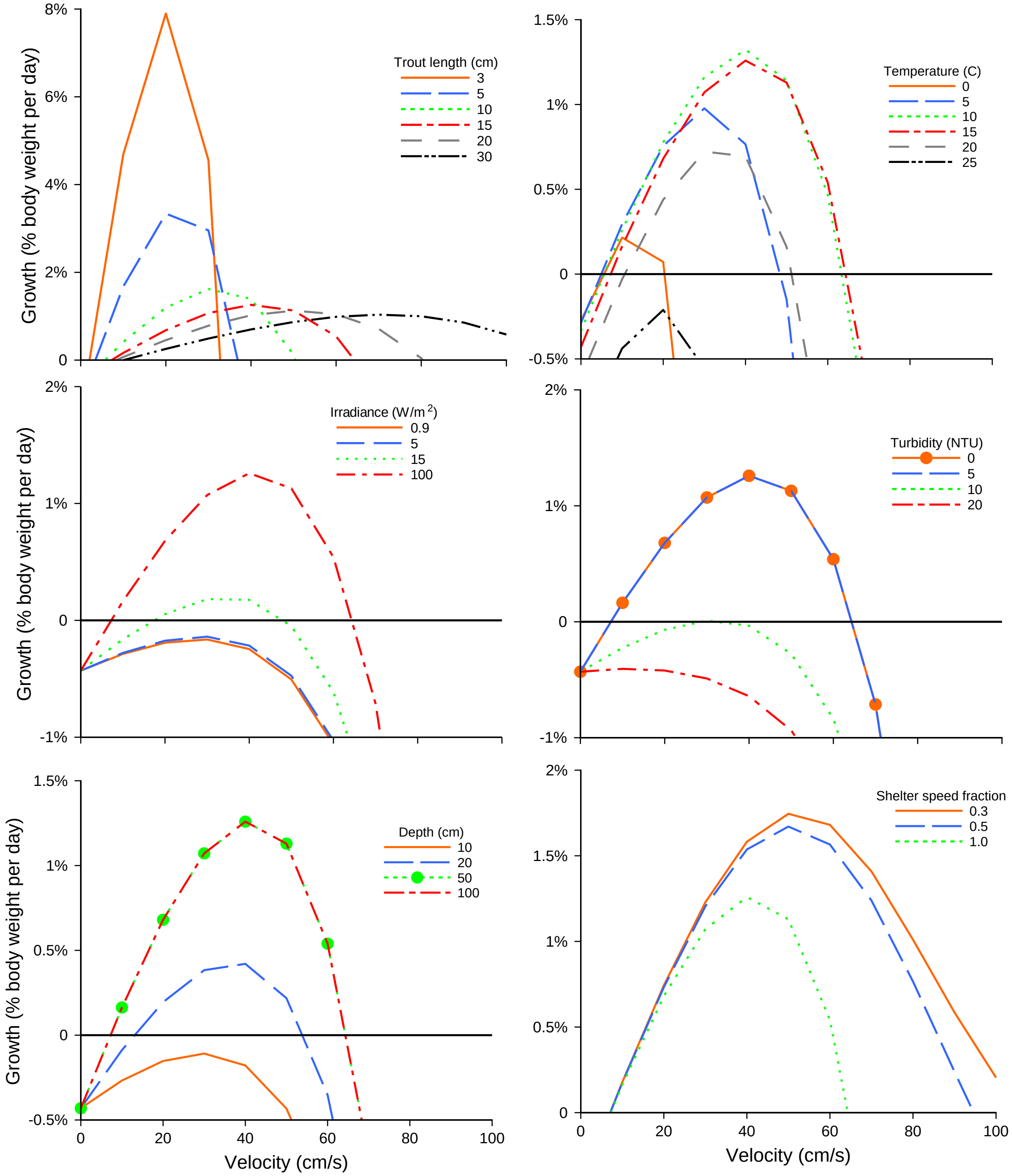


Figure 31. Exploration of drift feeding growth rates. Each panel shows how the relation between cell velocity (X) and trout growth rate (Y) depends on one other variable. Growth was calculated using the parameter values in Table 12 and Table 14, and a value of 3.0E-10 g/cm^3 for *reach-drift-conc*. Except as indicated in each panel's legend, simulations used these values: *trout-length,* 15 cm; *temperature*, 15°C; *cell-light*, 100 W/m^2 (daytime); *turbidity*, 0.0 NTU; *cell-depth*, 100 cm; and *reach-shelter-speed-frac*,1.0 (no velocity shelter).

9.23 Search feeding

Actively searching for benthic or drop-in food (Nakano et al. 1999) is an alternative to the drift-feeding strategy[49]. Our empirical research has shown that search feeding can be important when drift intake is low due to habitat conditions. Both model results (Harvey and Railsback 2009) and laboratory studies (Harvey and White 2008) indicate that search feeding is especially important when high turbidity makes drift feeding unprofitable. In simulating a field experiment with trout confined to stream reaches where flow was artificially reduced, Harvey and Railsback (2014) found that inSTREAM better matched observed growth when search feeding was included, presumably because there can be little drift food at very low velocities. Similarly, an application of inSTREAM in Iran was based on the assumption that trout in the Elarm River are especially dependent on search feeding because this river's low depth reduces drift intake (Hajiesmaeili 2019).

Like drift intake, search intake is assumed to be potentially limited by the search feeding intake rate, the physiological maximum intake rate, or the availability of search food. The overall submodel is:

$$I_S = min\left(C_S, C_{stepmax}, cellAvailableSearch\right)$$

where I_S is the daily rate of search food intake (g/d), C_S is the modeled rate of search food capture (g/d), $C_{stepmax}$ is the maximum consumption rate for the time step (Sect. 9.24), and *cell-available-search* is the rate (g/d) at which search food is available in the cell at the current time, which depends on the food production rate and depletion by larger fish (Sect. 9.11). (If the trout is a superindividual, the value of *cell-available-search* is divided by its value of *trout-superind-rep* so it represents availability per fish.) The remainder of this section describes the model of search capture rate C_S.

In contrast to drift feeding, no established models address search feeding by trout. The process is presumably complex due to, e.g., the influence of substrate type and prey taxa on prey availability. Given the uncertainties and complexities, inSTREAM takes a simple approach. We assume that the rate of search food intake is proportional to the rate at which search food becomes available: every trout searches for food at about the same rate, so intake increases linearly with food production. Search feeding intake is also assumed to decrease linearly to zero as water velocity increases to the trout's maximum sustainable swim speed. This velocity term represents how the ability of a fish to see and search for food decreases with velocity. (It does not represent the energetic cost of swimming at high velocities, which is considered in the respiration submodel; Sect. 9.25.)

We assume no negative effects of low light intensity or high turbidity on search feeding intake rate. This assumption was made in part to keep the submodel simple in the absence of literature on such effects. But it also seems reasonable to assume that search-feeding fish are typically closer to the bottom than their reaction distance (as modeled in Sect. 9.22.1), making feeding less sensitive to reaction distance. This assumption makes search feeding more beneficial relative to drift feeding at night, which corresponds with our anecdotal field observations that (in a clear stream) trout feeding at night appear usually to be bottom-oriented instead of drift-oriented.

The model of search food intake is:

[49] Search feeding intake is coded in the procedure `search-intake-for`.

$$C_S = reachSearchProd \times troutSearchArea \times max\left(\frac{troutMaxSpeed - cellVelocity}{troutMaxSpeed}, 0\right).$$

In this equation, ***reach-search-prod*** is a reach parameter representing the rate of search food production (g/d/cm^2; Sect. 9.11.2). The trout parameter ***trout-search-area*** (cm^2) can be interpreted as the area over which the search food production is consumed by one trout. *Trout-max-speed* is the trout's maximum sustainable swimming speed (Sect. 9.26). Typical values of ***reach-search-prod*** and ***trout-search-area*** are 1.0E-5 and 20,000, but ***reach-search-prod*** is often addressed in calibration (Sect 24).

9.24 Maximum food intake

The maximum food intake submodel determines the maximum rate (g/d; typically referred to as C_{max} in the fish bioenergetics literature) at which trout can consume food via either the search or drift feeding strategies[50]. Unlike many other applications of standard fish bioenergetics models (Hanson et al. 1997), C_{max} normally has no effect on food intake in inSTREAM; it is used only to limit intake under conditions—most commonly, extremely cold temperatures (e.g., Cunjak et al. 1998) but also when food availability is extremely high for small trout—when food processing and energy assimilation become limited by digestion rate instead of food intake rate. Unfortunately, C_{max} is poorly defined and difficult to measure, largely because it varies with factors such as the fish's exercise condition, food type, and feeding conditions in the laboratory (PG&E 1994, Myrick 1998).

Maximum food consumption rate is also time-scale dependent: fish (like other animals) can consume food at much higher rates over short periods than they could maintain over longer periods. Even though inSTREAM 7 uses time steps less than 1 day, we treat C_{max} as maximum consumption rate at a one-day time scale because conditions causing it to limit food intake are likely to persist for more than one time step. Another important reason to treat C_{max} as a daily rate is that the literature in which C_{max} has been measured at an explicit time scale has used 1.0 day (e.g., Mesa et al. 2012, Kepler et al. 2014). We do this by limiting food intake each time step so that the total intake over the 1.0-d period that ends at the end of the time step is less than the value of C_{max} for the current time step. Scaling C_{max} to a full day has potentially important implications because it could increase the ability of trout to maintain condition while hiding as much as possible: hiding during part of a day increases the maximum rate at which trout can feed during remaining phases of the same day, while eating at a high rate could sometimes cause C_{max} to limit consumption later in the day and hence make hiding more likely to be the best activity. This approach also greatly increases the rate at which simulated trout can consume food during time steps when feeding is especially profitable.

C_{max} is scaled to 1.0 d in several steps. First, the food consumption (g) in each time step is recorded as one of the trout's memory variables described in Sect. 9.12. Then the maximum rate at which a fish can consume food at the current time step ($C_{stepmax}$, g/d) is calculated as the daily maximum consumption rate C_{max} minus the consumption so far in the previous time steps of the 1.0-d period that ends at the end of the current time step, divided by *step-length*. Negative

[50] The maximum food intake submodel is coded in the procedure `c-stepmax-for`. However, to increase execution speed, the allometric weight term for C_{max} is coded in the procedure `update-trout` and the temperature term is updated in `update-habitat`.

$C_{stepmax}$ values (which result from declining temperatures making C_{max} less than on previous time steps) are set to 0.0.

Published equations for C_{max} (Hanson et al. 1997) include (a) an allometric function relating C_{max} to fish size, and (b) a temperature function. The equation used in inSTREAM is:

$$C_{max} = troutCmaxA \times troutWeight^{troutCmaxB} \times CmaxTemperatureFunction$$

where ***trout-cmax-A*** and ***trout-cmax-B*** are trout parameters. (Previous versions of inSTREAM defined ***trout-cmax-B*** as the value used here minus 1.0.) This equation is widely used with the allometric parameter values developed by Rand et al. (1993) for Rainbow Trout (Table 15) to model C_{max} of salmonids in general (e.g., Van Winkle et al. 1996, Railsback and Rose 1999, Booker et al. 2004).

The temperature function in the equation for C_{max} is based in part on laboratory studies on Rainbow Trout by Myrick (1998) and Myrick and Cech (2000). These studies focused on higher temperatures, measuring C_{max} at 10, 14, 19, 22, and 25º. Previous models of C_{max} for salmonids (Rand et al. 1993) used temperature functions based on the laboratory studies of From and Rasmussen (1984), who studied Rainbow Trout at temperatures of 5-22º, and of Elliott (1982), who studied Brown Trout.

Instead of an equation, the C_{max} temperature function for inSTREAM uses a set of points defining a piecewise-linear function. The value of *cmax-temp-function* for the temperature of the trout's reach is interpolated linearly from this function (Figure 32) ; any number of points can be used in this function. The interpolation method produces values falling on the line segments defined by the parameter points. (Temperatures below the lowest temperature of the points that define *cmax-temp-function* cause an error; temperatures above the highest point on the function receive a C_{max} value equal to that of the highest point.)

While several sets of equations and parameters for C_{max} and its temperature function have been published for different salmonid species, scrutiny of these publications indicates that differences among models of C_{max} are as likely the result of differences in experimental methods as inherent differences among species or stocks. Considering the inherent uncertainty in C_{max} and its limited effect on results of inSTREAM, we cautiously recommend values of 0.628 for ***trout-cmax-A*** and 0.7 for ***trout-cmax-B***, and the *cmax-temp-function* values in Table 15 for stream trout in general. However, more recent measurements by Mesa et al. (2012; bull trout eating fish prey) and Kepler et al. (2014; lake trout eating invertebrate larvae) are incompatible with the temperature function of Figure 32; they both found consumption rate to peak at temperatures below 20º. Users are advised to consider such literature when selecting parameter values for cold-adapted species.

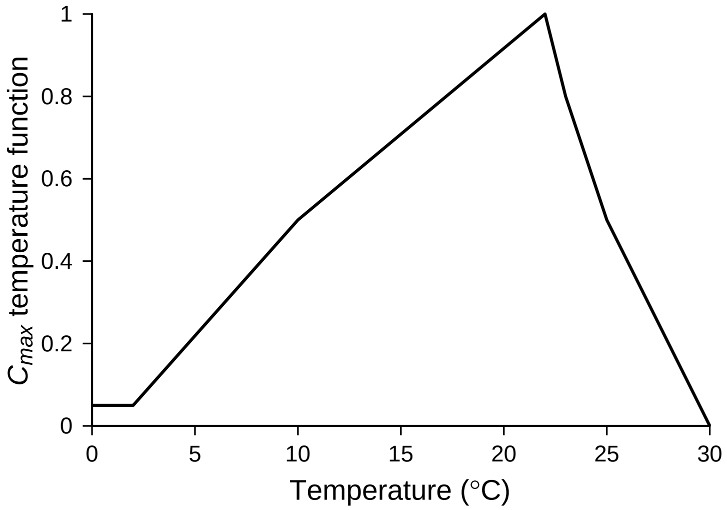


Figure 32: Temperature function for maximum food intake, with parameter values from Table 15.

Table 15: Example temperature function for maximum food intake, *cmax-temp-function*.

Temperature (°C)	**Function value (unitless)**
0.0	0.05
2.0	0.05
10	0.5
22	1.0
23	0.8
25	0.5
30	0.0

9.25 Respiration costs

This submodel calculates the rate at which trout lose energy, to be balanced against the energy intake rate in calculating growth[51]. All energy losses except those due to growth are categorized as "respiration" and (as an approximation) assumed to be represented by the standard technique of measuring respiration rate as oxygen consumption. The respiration formulation of inSTREAM is based on the "Wisconsin" bioenergetics formulation (Hanson et al. 1997; Deslauriers et al. 2017), which is very simple yet useful for approximating growth in fish population models. Parts of the respiration formulation of inSTREAM 7 differ from previous versions of inSTREAM and from the Wisconsin formulation, and parameter definitions and values have changed.

[51] Respiration is coded in the procedure `respiration-for`. However, the weight term is updated once per time step in procedure `update-trout` and the temperature term is updated once per time step in `update-habitat`.

InSTREAM neglects the egestion and excretion components of the Wisconsin model because they are relatively small and especially uncertain.

Respiration is treated as a function of body mass, water temperature, and swimming activity. The general formulation from Hanson et al. (1997) is:

$$troutRespiration = respMassTerm \times respTemperatureFunction \times respActivityFunction$$

where *trout-respiration* is the respiration (energy loss) rate (J/d), *resp-mass-term* relates "resting" metabolism (J/d; at 0° with no activity) to body mass, *resp-temperature-function* is a unitless function that represents how respiration increases with temperature, and *resp-activity-function* (unitless) represents how respiration increases with swimming speed. (A trout's swimming speed is determined in the habitat and activity selection submodel; Sect. 9.13.)

Resting metabolism is calculated as an allometric function:

$$respMassTerm = troutRespA \times troutWeight^{troutRespB}$$

where ***trout-resp-A*** (J/d/g) and ***trout-resp-B*** (unitless) are trout parameters with values taken from the bioenergetics literature. The parameters that Rand et al. (1993; summarized in Appendix A of Hanson et al. 1997) developed for Steelhead Trout are widely used for stream trout in general. For the multiplier, Rand et al. used a value of 0.00264 g oxygen per g fish per day, and an exponent of -0.217. Converting the units with an oxycalorific constant of 13,560 J/g O_2 (Deslauriers et al. 2017) and converting from a mass-specific form (J/d/g) to the absolute form (J/d) results in values of 36 J/d/g for ***trout-resp-A*** and 0.783 for ***trout-resp-B***.

The temperature function of inSTREAM 7 is revised in accordance with recent, more comprehensive data. The temperature function of the Wisconsin model and previous versions of inSTREAM used an exponential function:

$$respTemperatureFunction = exp(troutRespC \times temperature).$$

This function was used by Rand et al. (1993) with a value of 0.0682 $^{\circ}C^{-1}$ for the parameter ***trout-resp-C***; it produced a doubling (98% increase) in respiration between 10 and 20° and an increase of 40% between 20 and 25°. Rand et al. (1993) based this parameter value on observations of only two temperatures by Rao (1968).

Several more comprehensive data sets have been generated using field respirometry on wild fish. Verhille et al. (2016) measured resting metabolic rate (*RMR*), as an oxygen consumption rate, for two trout at each of 13 temperatures between 14 and 25ºC. They used a field swim chamber and respirometer and observed wild Rainbow Trout of ~13 cm length. (They concluded that their study population, from the lower Tuolumne River of California's Central Valley, may be physiologically adapted to high temperatures.) Their observations did not exhibit high variability among individual fish. They fit a quadratic equation to their data; the equation fits their data closely over the range of observed temperatures but produces unrealistic values at lower temperatures.

A similar study was conducted by Lee et al. (2003), but on adult salmon during their spawning runs. That study measured *RMR* as "routine" oxygen consumption at a low swimming speed. Measurements were taken at ambient temperatures from 5-20°, but this range was extended to 23° by "adjusting" some test fish by holding them at higher than ambient temperature. The results were reported as exponential equations fit to the rather highly variable data (*RMR* varied among individuals by as much as a factor of two). Separate equations were fit with and without the "adjusted" fish. (The equations were misprinted in the original publication and republished in a correction.)

The results of these two studies fit exponential functions of the type used in the Wisconsin model relatively well, but exponential functions fail to capture the steep increase in respiration at temperatures > 20°, especially in the adult salmon data (Figure 33). This steep increase appears due to physiological breakdown at high temperatures (Eliason et al. 2013a) and is likely to have strong effects on model results at high temperatures (inSTREAM results are, in general, quite sensitive to respiration rates; Railsback et al. 2009). (Eliason et al. 2013a report other data on adult salmon also showing a steep increase in RMR at 25-26º.)

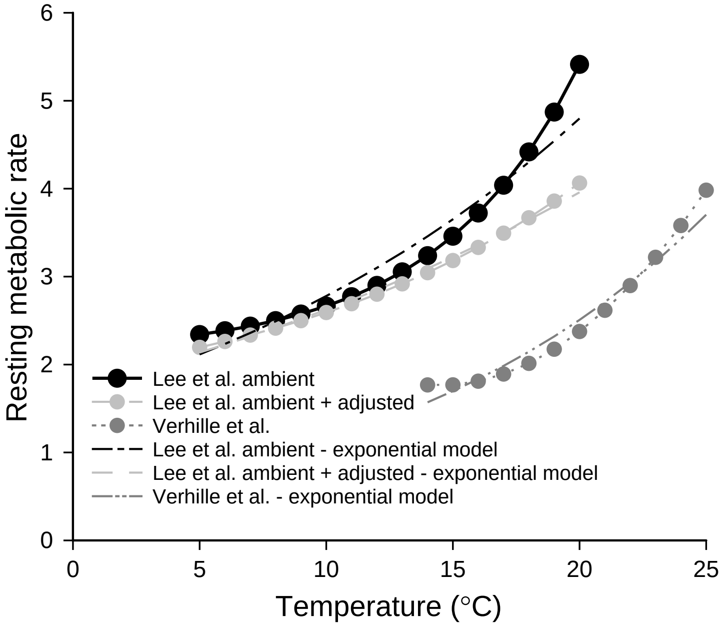


Figure 33: Resting metabolic rate as a function of temperature in the studies of Lee et al. (2003) and Verhille et al. (2016). Curves with round symbols show the equations developed in those publications to fit the data; curves labeled "exponential model" are exponential functions fit to the other curves over the range of observed temperatures. The two data sets of Lee et al. are without and with fish adjusted to higher than ambient temperatures.

To capture the steep increase in *RMR* at high temperatures we replace the Wisconsin model's exponential function with an exponential function of the temperature squared:

$$respTemperatureFunction = exp(troutRespC \times temperature^2).$$

This function fits the data more closely at high temperatures and produces little response to temperature below 5° (Figure 34). The value of ***trout-resp-C*** is simply the exponent found via exponential regression of the observed *RMR* values to *temperature*^{2}. The curves shown in Figure

34 have values of ***trout-resp-C*** of 0.0022ºC^{-2} (Lee et al., ambient fish only), 0.0016ºC^{-2} (Lee et al., ambient and adjusted fish), and 0.0020ºC^{-2} (Verhille et al.). Small changes in the value of this parameter have strong effects. Given values of *resp-temperature-function* between 1.5 and 3.0 at 20º, ***trout-resp-C*** should fall within the range of 0.0011 to 0.0027.

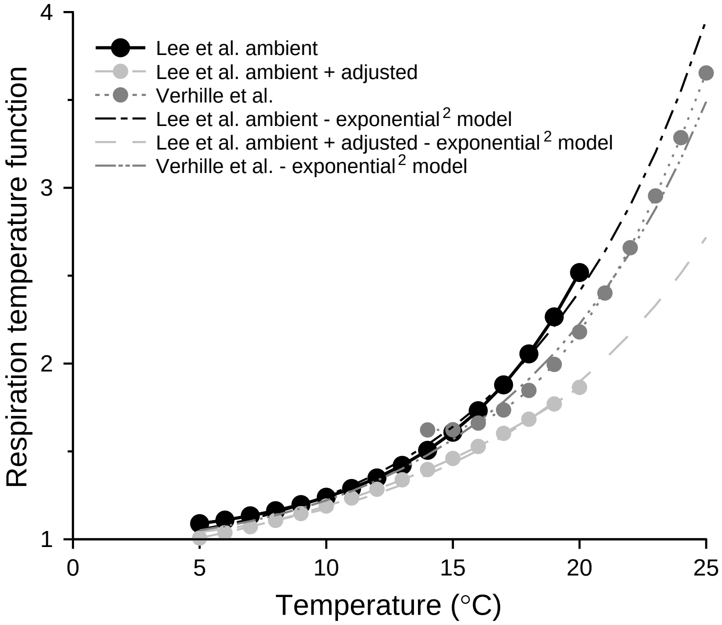


Figure 34: Respiration temperature functions fit to observations of Lee et al. (2003) and Verhille et al. (2016) using an exponential function of the square of temperature (the "exponential^2 model"). The curves with round symbols show the equations fit to data in these publications, transformed into the temperature function by dividing their values by respiration at 0 from the exponential^2 model. The curves without symbols are the exponential^2 models.

The swimming activity function for inSTREAM 7 is also new. Previous versions used the Wisconsin model formulation, which assumes the function is an exponential function of swimming speed. That formulation does not account for how swimming costs vary with trout body size: using it, swimming at (e.g.) 50 cm/s increases respiration by the same percentage whether a trout is 5 cm long (for which 50 cm/s would be extremely energetic) or 50 cm long (for which 50 cm/s would be only moderately energetic).

Our solution to this problem is to scale swimming speed as a fraction of *trout-max-speed* (Sect. 9.26). (Whiterod et al. 2013 also addressed this problem and developed a similar approach, scaling swimming speed by body length.) We found three relevant data sets in the literature. (Data for adult salmon have also been published by, e.g., Eliason et al. 2013b, MacNutt et al. 2006.) Gallaugher et al. (2001) measured oxygen consumption by 30-cm exercise-conditioned Chinook Salmon over a range of swimming speeds reported as a percentage of U_{crit} (the measure of swimming capacity used as the basis of *trout-max-speed*; Sect. 9.26). Tudorache et al. (2008) made similar measurements for 8-cm Brown Trout. For each of these data sets, we converted swimming speed from fraction of U_{crit} to fraction of *trout-max-speed* by dividing by 0.9 (we calculate *trout-max-speed* as 0.9×U_{crit}; Sect. 9.26). Johansen et al. (2020) measured oxygen consumption by 33-cm Rainbow Trout at water velocities from zero to 100 cm/s (measurements of trout not using a velocity shelter), and fit a statistical model to the results; we calculated a maximum sustainable swimming speed of 113 cm/s from our submodel (Sect. 9.26). The trout used by Johansen et al. (2020) were from a hatchery and apparently not exercise-conditioned.

The data of Gallaugher et al. (2001), Johansen et al. (2020), and the other data for adult salmon cited above, show a steep increase in measured oxygen consumption as swimming speed approaches and exceeds *trout-max-speed*. This steep increase appears to result from trout switching to less-sustainable swimming gaits and metabolic processes (Eliason et al. 2013b). The Tudorache et al. (2008) data fit an exponential model well, but to capture the steep increase observed in other data sets we use an exponential function of the square of swimming speed divided by *trout-max-speed* (Figure 35):

$$respActivityFunction = exp(troutRespD \times (troutSwimSpeed/troutMaxSpeed)^2).$$

The trout data of Tudorache et al. (2008) produce a value of 0.90 for ***trout-resp-D***, the trout data of Johansen et al. (2020) produce a value of 2.4, and the Chinook Salmon data of Gallaugher et al. (2001) produce a value of 1.60 (all estimated via exponential regression of oxygen consumption vs. the square of swimming speed divided by *trout-max-speed*, with the regression intercept set to 1.0). The value of 0.90 appears to be low considering observations, such as those of Eliason et al. (2013b; adult Sockeye Salmon), that oxygen consumption rises to 4-5 times resting levels during swimming near U_{crit}. The value of 2.4 from Johansen et al. is considerably higher than from the other datasets, perhaps reflecting the use of hatchery trout. If we assume that resident trout are more efficient swimmers than adult salmon, *resp-activity-function* values between 3 and 5 seem reasonable for swimming at *trout-max-speed*; such values result from ***trout-resp-D*** in the range of 1.1 to 1.6.

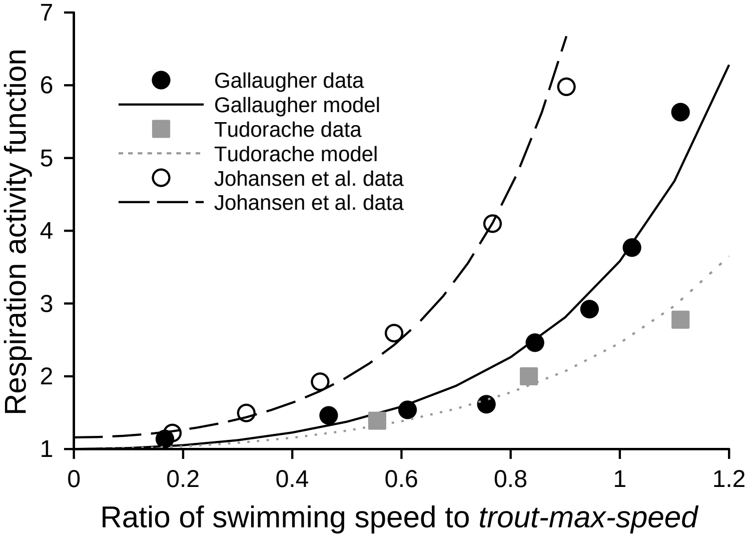


Figure 35: Relation between swimming speed and oxygen consumption, translated into the activity function for respiration costs, for 30-cm Chinook Salmon (Gallaugher et al. 2001), 8-cm Brown Trout (Tudorache et al. 2008), and 33-cm Rainbow Trout (Johansen et al. 2020). The curves represent an exponential regression fit to the square of swimming speed divided by *trout-max-speed*.

An exploration of the respiration submodel (Figure 36) indicates that total daily respiration cost varies strongly with all three driving variables: fish size, temperature, and swimming speed. The sharp increase in respiration at low temperatures and moderate swimming speed (top left of the contour plots) is due to the use of *trout-max-speed* in the activity function. Maximum swimming speed is low at low temperatures (Sect. 9.26), making the ratio of swimming speed to *trout-max-speed* high. Whether or not this result is realistic, it is unlikely to be important because the sharp increase in respiration happens only at swimming speeds approaching *trout-max-speed* (which is, at 2° and with the standard parameter values, 18 cm/s for a 5-cm trout and 41 cm/s for a 20-cm trout).

Finally, the *resp-activity-function* equation produces numerical problems when the ratio of swimming speed to *trout-max-speed* is high: when this ratio is above ~25, the equation produces values larger than NetLogo software can handle. (Such values occur during habitat selection when model trout evaluate cells with extremely high velocities.) To avoid this problem, inSTREAM 7 sets the value of *trout-respiration* to an arbitrary large number (999,999) when the swim speed ratio exceeds 20.

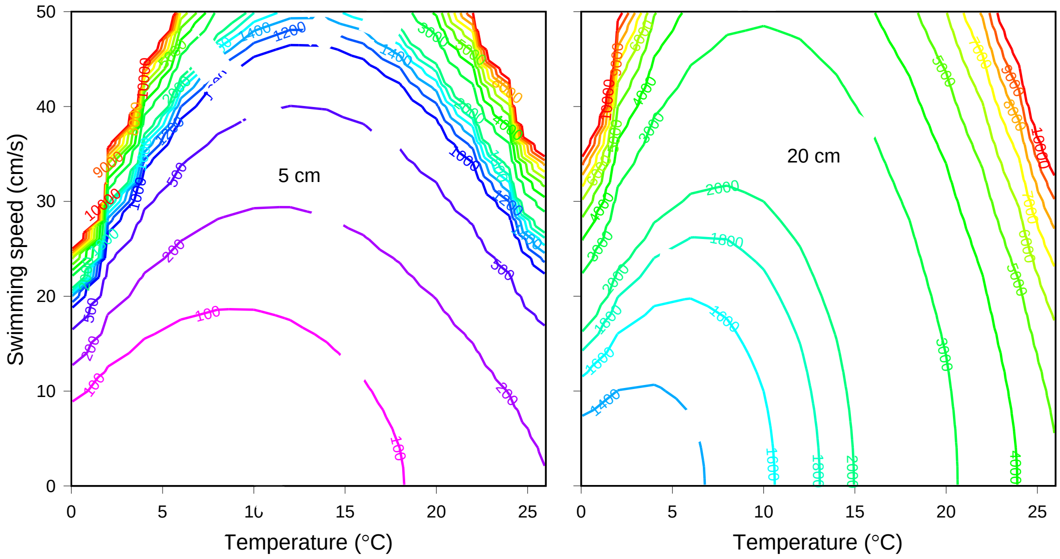


Figure 36. Respiration submodel results as a function of temperature and swimming speed, using the parameter values of Table 16. Results are for healthy 5-cm (left) and 20-cm (right) trout. Contours indicate simulated total respiration (J/d).

Table 16: Parameters and values for respiration costs.

Parameter	Definition and units	Basis	Standard value
trout-resp-A	Multiplier in allometric term for *resp-mass-term* (J/d/g)	Rand et al. (1993)	36

Parameter	Definition and units	Basis	Standard value
trout-resp-B	Exponent in allometric term for *resp-mass-term* (unitless)	Rand et al. (1993)	0.783
trout-resp-C	Parameter in the temperature function (°C^{-2})	Verhille et al. (2016)	0.0020
trout-resp-D	Parameter in the activity function (unitless)	Gallaugher et al. (2001), Tudorache et al. (2008)	1.4

9.26 Maximum sustainable swimming speed

This submodel updates the trout state variable *trout-max-speed*, maximum sustainable swim speed (cm/s)[52]. This state variable is a component of both kinds of feeding, so it strongly affects the relationship between a cell's velocity and habitat quality. This variable represents the highest speed that a trout can maintain for long periods—for feeding throughout a time step—so it must be a speed that fish can swim for hours, not a burst or short-term maximum speed. The formulation for *trout-max-speed* is based on literature values of "critical swimming speed" (U_{crit}), a standard approach to estimating maximum sustainable speed in a laboratory test chamber. Measurement of U_{crit} involves repeatedly stepping up the swimming speed and holding it for a specified time interval, until the fish is exhausted; different time intervals can be used to estimate short-term vs. long-term sustainable swim speeds. We used long-term values of U_{crit} to represent *trout-max-speed*. Myrick (1998) and Farrell (2007) indicate that trout may start to use white (fast-twitch) muscle fibers at 90-95 percent of U_{crit}. Therefore, a better estimate of the speed fish can sustain for long periods is 90 percent of the U_{crit} (C. Myrick, Department of Fish, Wildlife, and Conservation Ecology, University of California, Davis, pers. comm. with S. Railsback, 10 May 1999).

U_{crit} for trout has been measured at different temperatures and fish lengths by a number of researchers. These studies examined brown (Butler et al. 1992), cutthroat (Hawkins and Quinn 1996, MacNutt et al. 2004), and rainbow and golden trout (Schneider and Connors 1982; Taylor et al. 1996; Alsop and Wood 1997; Myrick 1998, also published in Myrick and Cech 2000; Myrick and Cech 2003). [The study by Griffiths and Alderdice (1972) was not used even though it has been the basis of several previous models of maximum swimming speed. Griffiths and Alderdice measured juvenile Coho Salmon swimming speed over temperatures between 2 and 26° C. However, they did not provide sufficient information to distinguish the effects of fish size and temperature, and apparently did not control these two variables separately.] In addition, a number of measurements have been made of juvenile salmon (Brauner et al. 1992, 1994) and on adult salmon on their spawning migrations (e.g., Farrel et al. 2003; Lee et al. 2003; MacNutt et al. 2006).

[52] Sustainable swim speed is coded in the procedure `max-swim-speed-for`. However, the length term is updated once per time step in procedure `update-trout` and the temperature term is updated once per time step in `update-habitat`.

There is considerable variability among these studies, likely due in part to differences in experimental equipment and techniques and to variability in the exercise condition of the fish. However, two general conclusions can be drawn. First, *trout-max-speed* increases with fish length (Figure 37). Second, *trout-max-speed* varies nonlinearly with temperature, peaking at temperatures around 10-15° (Figure 38).

The formulation for trout-max-speed therefore has two terms: the first represents how swimming speed at 10-15° varies with fish length, and the second modifies trout-max-speed for temperature. Therefore, we model maximum swim speed as:

$$troutMaxSpeed = (MSP_A \times troutLength + MSP_B) \times (MSP_C \times T^2 + MSP_D \times T + MSP_E)$$

where T is *temperature* and MSP_A through MSP_E are the parameters ***trout-max-speed-A*** through ***trout-max-speed-E***. The parameters ***trout-max-speed-A*** and ***trout-max-speed-B*** define the linear relation between trout length and maximum speed; the values for these parameters in Table 17 were estimated via linear regression on the trout observations shown in Figure 37. The parameters ***trout-max-speed-C*** through ***trout-max-speed-E*** define a quadratic function of temperature; the parameter values were fit to the observations in Figure 38.

Figure 37 indicates that the parameters developed here for trout would highly overestimate swim speeds for adult salmon. However, Lee et al. (2003) and MacNutt et al. (2006) produced data similar to those in Figure 37 for adult salmon during their spawning migrations; those data produce a similar humped response of U_{crit} to temperature, with peaks around 15-17°.

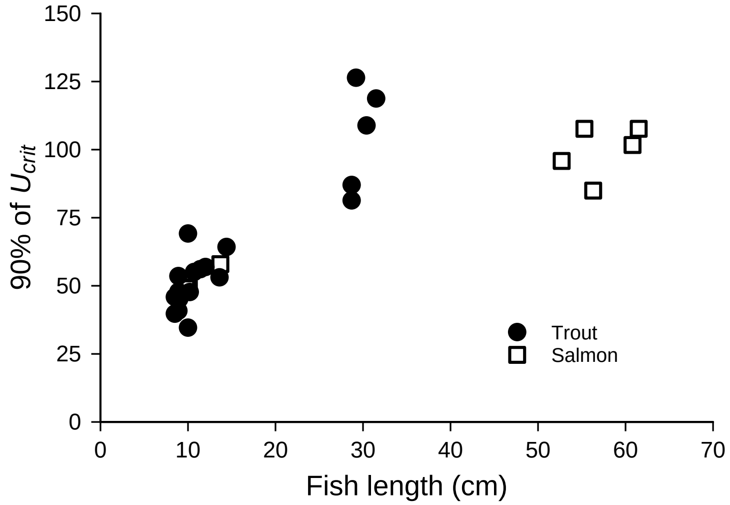


Figure 37: Maximum sustainable swim speed, defined as 90% of observed U_{crit}, as a function of fish length. Values are from measurements made at temperatures between 9.5 and 15°. Round symbols are observations of resident trout from the sources cited in Figure 38. Square symbols are observations of salmon parr and smolts (9-14 cm length) by Brauner et al. (1992, 1994) and of salmon adults (>50 cm length) by Farrell et al. (2003).

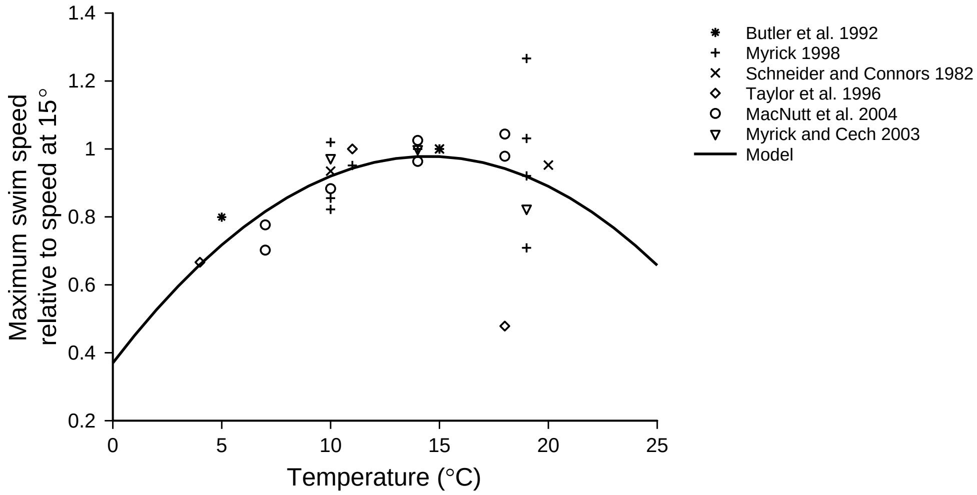


Figure 38: Variation in maximum sustainable swim speed with temperature. The Y value is measured maximum speed divided by speed measured at or near 15° in the same study. Observations from six studies are shown separately. The curve illustrates the model's temperature function fit to these data.

Table 17: Parameters and values for maximum sustainable swim speed

Parameter name	Definition and units	Basis	Standard value
trout-max-speed-A	Length coefficient in relation between trout length and maximum swim speed (s^{-1})	Regression fit to observations of resident trout (Figure 37)	2.8
trout-max-speed-B	Constant in relation between trout length and maximum swim speed (cm/s)	Regression fit to observations of resident trout (Figure 37)	21.0
trout-max-speed-C	Temperature squared coefficient in temperature effect on maximum swim speed (°C^{-2})	Regression fit to observations of resident trout (Figure 38)	-0.0029
trout-max-speed-D	Temperature coefficient in temperature effect on maximum swim speed (°C^{-1})	Regression fit to observations of resident trout (Figure 38)	0.084

Parameter name	Definition and units	Basis	Standard value
trout-max-speed-E	Constant in temperature effect on maximum swim speed (unitless)	Regression fit to observations of resident trout (Figure 38)	0.37

9.27 Spawning readiness

Female trout (those with *trout-sex* = "female") execute this submodel once per day to determine whether they will spawn that day[53]. (Male participation in spawning is described in Sect. 9.30.) Salmonids are clearly capable of adapting some of their reproductive behaviors to environmental conditions and their own state, especially by deciding whether or when to spawn each year considering their current size and condition and habitat conditions (e.g., Nelson et al. 1987). However, inSTREAM's objectives do not justify a detailed representation of such processes as the bioenergetics of spawning or the adaptive decision of whether to spawn each year considering the fish's current state and expected growth and mortality risks. Instead, inSTREAM's spawning methods simply force model trout to reproduce general spawning behaviors observed in real trout. Spawning-related behaviors and mechanisms are included only if they appear important for simulating flow and temperature effects on reproduction or for representing the effects of spawning on the adult spawners.

On the daytime phase of each day, each female trout determines whether it meets all of the fish- and habitat-based spawning criteria described below. These spawning criteria limit spawners to females of adequate size and physiological condition; and restrict spawning to physical conditions (dates, flows, temperatures) when spawning has been observed in real trout, presumably because spawning is more likely to be successful under those conditions. The criteria for readiness to spawn do not include the availability of good spawning habitat; we assume trout will spawn whether or not ideal gravel spawning habitat is present. Observations reported by Magee et al. (1996) support that assumption.

On the days when all the spawning criteria are met for a female, a random Bernoulli trial determines whether she actually spawns; the probability of spawning is the trout parameter ***trout-spawn-prob*** (unitless). This stochastic selection of spawning date imposes some variability in when individual fish spawn, which can be important to the population's reproductive success. In some situations, flow fluctuations during the spawning season can scour or desiccate redds during short periods. Variation in when fish spawn can keep such events from causing unnatural population-level reproductive failure. Imposing variability in spawning times can also be necessary to reproduce observations at some sites, e.g., more variation in spawn timing where physical habitat is more diverse (Jonsson and Jonsson 2011). The value of ***trout-spawn-prob*** also gives the model user some control over what percent of spawning-sized fish actually spawn. If the inverse of ***trout-spawn-prob*** is large compared to the number of days in the spawning period (e.g., 1 / ***trout-spawn-prob*** is greater than the number of potential spawning days), then it is likely that some potential spawners will not spawn. A typical value of 0.04 causes an average of

[53] Spawning readiness is programmed in the procedure `ready-to-spawn?`. However, the random draw against ***trout-spawn-prob*** is in the procedure `spawn`.

25 percent of ready fish to spawn in the first week of suitable conditions and 68 percent to spawn in 28 days of suitable conditions.

The following subsections describe the spawning criteria, with example parameter values in Table 18. None of these criteria are well defined in the literature because trout spawning is difficult to observe. The criteria include characteristics of both the potential spawner and its habitat reach. The criteria based on flow and temperature are evaluated using the habitat reach currently occupied by the potential spawner; inSTREAM does not represent migration to different reaches in search of better spawning conditions.

9.27.1 Minimum age, length, and condition

Because inSTREAM does not explicitly simulate the bioenergetics of reproduction, it uses trout length, age, and condition to predict energetic readiness to spawn. Minimum values of these characteristics are used to ensure that only fish with energy reserves comparable to those needed for gonad production can actually spawn. The habitat and activity selection submodel provides no incentive for model trout to gain size or condition in preparation for spawning, so this submodel should not demand that females have extremely high condition to spawn. Length and condition are the primary indicators of spawning readiness as they are related to energy reserves, but the age minimum is useful in model runs where fish growth and condition are not well calibrated. The length and age criteria are also important for imposing site-specific variation in spawning size and age that, in real fish, could be local adaptations (behavioral or genetic) to habitat and competitive conditions. These criteria simplify many complex factors that drive the age and size at which salmonids spawn.

Female trout in inSTREAM cannot spawn unless their age (*trout-age*) equals or exceeds the value of the parameter ***trout-spawn-min-age***, an integer age in years. Trout also cannot spawn unless their length (*trout-length*) exceeds the parameter ***trout-spawn-min-length*** (cm). (This parameter is also a key variable in the fitness measure used as a basis of habitat and activity selection decisions; Sect. 9.13.2.) Finally, for a trout to spawn, its condition factor *trout-condition* must exceed the parameter ***trout-spawn-min-cond*** (unitless).

Values for ***trout-spawn-min-age*** and ***trout-spawn-min-length*** can vary considerably among sites and can often be estimated from site-specific census data. For Cutthroat Trout in the relatively small, infertile Little Jones Creek, Railsback and Harvey (2001) used 1 y for ***trout-spawn-min-age***: field observations indicated that spawning in age 1 trout occurs, if rarely. Rosenberger et al. (2015) also documented spawning by age-1 in Rainbow Trout in small streams providing rapid growth. Railsback and Harvey (2001) based a value of 12 cm for ***trout-spawn-min-length*** on their field observations and studies of similar sites. In contrast, Meyer et al. (2003) found that trout in the large South Fork Snake River did not mature until they were 30 cm long and five years old, and provide data on how these spawning age and size parameters can vary with habitat conditions. Parra et al. (2014) modeled how these characteristics varied with altitude and latitude in European populations of Brown Trout. Carlson and Seamons (2008) review evidence indicating that these characteristics are genetically heritable yet likely to respond rapidly to environmental change.

Considering (a) the non-standard definition of condition factor (Sect. 9.20); (b) that the growth formulation causes *trout-condition* to be less than 1.0 on any time step a fish's energy intake is less than respiration costs, such as when hiding; and (c) that the bioenergetics of reproduction are not explicitly represented and trout have no incentive to put on weight in anticipation of spawning, a ***trout-spawn-min-cond*** value close to 0.98 is appropriate. Preliminary model results for condition can also be used to estimate a value that represents relatively successful individuals at the time of spawning.

9.27.2 Not spawned this season

Trout are assumed not to spawn more than once per annual spawning season. The trout (both males and females) in inSTREAM have a variable *trout-spawned-this-season?* set to FALSE on the first day of the spawning season (as defined in Sect. 9.27.3), and then set to TRUE when the trout spawns. Trout—females and males—cannot spawn if their value of *trout-spawned-this-season?* is TRUE.

9.27.3 Date window

Salmonids generally have distinct annual spawning seasons, which is not surprising because time of year is an important predictor of factors critical to successful spawning. For example, early spring spawning may make eggs and fry more vulnerable to cold temperatures or streambed scour from high flows from snowmelt, but spawning too late may make offspring more vulnerable to high temperatures and reduce their ability to compete with earlier-spawned juveniles. Therefore, in inSTREAM fish can spawn only on days within a user-specified date window.

Two trout parameters, ***trout-spawn-start-day*** and ***trout-spawn-end-day***, define the first and last dates on which spawning can occur. These parameters specify a month and day of the month (in M/d format) applicable to all years. (The spawning window can extend from the end of one year into the next; for example, ***trout-spawn-start-day*** can be 12/1 with ***trout-spawn-end-day*** 2/1.) Table 18 provides example values.

9.27.4 Temperature range

Temperature is widely accepted as a factor controlling the timing of spawning (e.g., Lam 1988). Temperature could be used by potential spawners as a cue for seasonal changes, and spawners likely avoid temperatures that induce egg mortality (sects. 9.32 and 9.33). Specifying the time scale at which temperature is evaluated is an important part of this criterion: do potential spawners consider only the temperature at the current time step, or temperatures over some previous time period? This question is especially important because inSTREAM 7 can use temperature input that includes within-day variation.

We model the temperature criterion explicitly at a daily time scale. A daily mean temperature is estimated by obtaining a list that contains all temperature input values associated with the 1.0-day period that ends at *sim-time* (the end of the current time step, which, for spawning, is in the afternoon before dusk starts), for the reach currently occupied by the trout. If that list is empty (e.g., because only weekly input is used), the current value of the reach variable *temperature* is added to it. When daily input is used in the recommended format, with each day's input associated with noon that day (Sect. 8.1), this list will contain only the current day's temperature. If input is provided at sub-daily intervals, this list can contain more values. If the input is provided at intervals longer than one day, this list could contain only the current temperature.

A trout can spawn only if the mean of the temperatures on this list is (a) equal to or greater than a trout parameter for minimum spawning temperature (***trout-spawn-min-temp***, °C) and (b) less than or equal to the parameter ***trout-spawn-max-temp*** (°C). We recommend that the values for these parameters reflect the temperatures assumed to cause egg mortality; Table 18 provides example values.

9.27.5 Flow limit

The flow limit criterion implements the assumption that fish will not spawn during high flow events. During unusually high flow, cells with depths and velocities suitable for redds (Sect. 9.29) are likely to be along river margins where redds would be at risk of dewatering when flows recede; and redds are vulnerable to scouring during high flows (Sect. 9.35).

The same time scale issues discussed above for temperature (Sect. 9.27.4) are also relevant for the flow limit. We use 1.0 day as the time scale: the criterion uses the maximum of all flows in the input file associated with times within a 1.0-day period that ends at *sim-time* and, if there are no such flows, it uses instead the current value of the reach variable *flow*. A trout is not allowed to spawn if this flow is greater than the value of reach parameter ***reach-max-spawn-flow*** (m^3/s). (This is a habitat parameter instead of a trout parameter because it varies among reaches.) This parameter is highly site-specific.

When multiple reaches are simulated, a trout can spawn in a different reach than the one it occupies when deciding whether to spawn. However, the flow limit is always evaluated using the flows and value of ***reach-max-spawn-flow*** of the currently occupied reach.

9.27.6 Steady flows

Trout are assumed not to spawn during unsteady flows because flow fluctuations place redds at risk of dewatering or scouring. We use a time scale of two days to evaluate this criterion. The trout parameter ***trout-spawn-max-flow-change*** (unitless) defines the maximum flow variability for spawning: if the maximum fractional change in flow—in the trout's current reach—over the current and previous day exceeds ***trout-spawn-max-flow-change*** then trout cannot spawn.

The maximum fractional change in flow is determined by creating a list containing all flow values in the input associated with the time interval between *sim-time* and 2.0 days before *sim-time*. With standard daily input associated with noon (Sect. 8.1), this list will have two values, from the current and previous days. If flow input is at intervals of greater than 1.0 day, the list may have only one value, for the current day. In that case, we assume that the input does not have sufficient resolution to evaluate flow steadiness so this criterion is ignored.

The maximum fractional change in flow is evaluated as:

$$fracFlowChange = \mathrm{abs}(Q_{max} - Q_{min}) / Q_{min}$$

where Q_{max} and Q_{min} are the maximum and minimum values on the list of flows and abs() is the absolute value function. Van Winkle et al. (1996) and Railsback and Harvey (2001) used 0.2 for ***trout-spawn-max-flow-change***.

Table 18: Example parameter values for spawning readiness

Parameter	Definition and units	Basis	Example values
trout-spawn-min-age	Minimum age (years) at which trout can spawn	Site-specific estimates	Small streams: 1-2 years Large rivers: 3-5 years

Parameter	Definition and units	Basis	Example values
trout-spawn-min-length	Minimum length (cm) at which trout can spawn	Site-specific estimates	Small streams: 10-15 cm Large rivers: 20-30 cm
trout-spawn-min-cond	Minimum value of *trout-condition* for female spawners (unitless)	Should be relatively high but consider that *trout-condition* is always < 1.0	0.98
trout-spawn-start-day	Day of the year that that spawning season begins (M/d)	Site- and species-specific estimates	Sierra Nevada Rainbow Trout (Van Winkle et al. 1996): 4/1 Sierra Nevada Brown Trout (Van Winkle et al. 1996): 10/1 Coastal Cutthroat Trout (Railsback and Harvey 2001): 4/1
trout-spawn-end-day	Day of the year (M/d) that is the last day of the spawning season	Site- and species-specific estimates	Sierra Nevada Rainbow Trout: 6/30 Sierra Nevada Brown Trout: 12/31 Coastal Cutthroat Trout: 5/31
trout-spawn-min-temp	Minimum temperature (on the current time step, in the trout's reach) at which spawning can occur (°C)	Create a spawning temperature range that excludes temperatures causing significant temperature mortality to eggs (sects. 9.32, 9.33)	Sierra Nevada Rainbow Trout (Van Winkle et al. 1996): 8 Sierra Nevada Brown Trout (Van Winkle et al. 1996): 4
trout-spawn-max-temp	Maximum temperature at which spawning can occur (°C)		Sierra Nevada Rainbow Trout (Van Winkle et al. 1996): 13 Sierra Nevada Brown Trout (Van Winkle et al. 1996): 10
trout-spawn-max-flow-change	Maximum flow change (as fraction of current flow) after which trout can spawn	Van Winkle et al. (1996), Railsback and Harvey (2001)	0.2

9.28 Spawning

A female trout that meets all criteria for spawning, including the Bernoulli trial against parameter ***trout-spawn-prob***, then spawns and creates a redd. (We refer here to such trout as "spawners".) Spawning includes five steps[54].

The first spawning step is selecting and moving to a spawning cell, using the spawning site selection submodel (Sect. 9.29). (The only effect this move has on the spawner is that when it next selects its habitat and activity it starts from the cell it spawned in.)

The second spawning step is creating a redd. A spawner creates a new redd with its location set to the spawning cell. The redd's state variables (Sect. 4.1.5) are initialized by setting *redd-species* to the spawner's species (its value of *trout-species*), *redd-frac-developed* to 0.0, and *redd-emerge-days* to 0 d. The initial value of *redd-num-eggs* depends on the spawner's fecundity (a function of length) and losses during spawning:

$$n = (mL^E) \times V$$

where n is the value given to the new redd's variable *redd-num-eggs*, m is the trout parameter ***trout-spawn-fecund-mult***, L is the spawner's length (*trout-length*), E is the parameter ***trout-spawn-fecund-exp***, and V is the parameter ***trout-spawn-egg-viability***[55].

The first term in this equation calculates the spawner's fecundity, the number of eggs it produces. Relations between fecundity and length are surprisingly variable even within species. Van Winkle et al. (1996) developed values of ***trout-spawn-fecund-mult*** and ***trout-spawn-fecund-exp*** for Brown Trout from Avery (1985). These values (Table 19, Brown Trout values) result in fecundities of 60 eggs for a small spawner of 12 cm and 220 eggs for a spawner of 20 cm, corresponding well with citations provided by Carlander (1969). Meyer et al. (2003) developed parameters for fecundity from 26 observations of resident Cutthroat Trout, with lengths between 10 and 30 cm. The total lengths reported by Meyer et al. were converted to fork length by applying a ratio of 0.97 (Carlander 1969). The resulting parameter values (Table 19, Cutthroat Trout values) produce fecundities approximately 50 percent higher than those of Van Winkle et al. (1996). Parra et al. (2014) evaluated variation in the fecundity parameters with altitude and latitude in European Brown Trout. The differences between the two parameter sets reported in Table 6 may be more a result of random variation or differences among sites than real differences among trout species.

The second term, ***trout-spawn-egg-viability***, represents the fraction of eggs successfully fertilized and placed in the redd. (Even though ***trout-spawn-egg-viability*** has the same effect mathematically as ***trout-spawn-fecund-mult***, fecundity and egg viability are treated separately to allow clear use of the extensive literature on fecundity.) The number of viable eggs in a redd can be considerably less than the female's fecundity if some eggs are washed away, incompletely buried, or eaten by other fish during redd creation; or if some are not fertilized. This parameter can also be used to represent mortality of eggs and alevins due to processes not explicitly included in the model's redd survival action. There is little published literature to support consistent values of ***trout-spawn-egg-viability*** for stream salmonids. For example, Healey (1991) found only a few, conflicting, studies on egg deposition and loss for Chinook Salmon. He concluded that egg loss is often low but can be high in high-velocity streams. Anecdotal evidence

[54] With the exceptions in other footnotes, spawning is coded in the procedure `spawn`.

[55] The number of viable eggs is calculated in `num-viable-eggs`.

from salmon and trout in coastal California suggests that the number of emerging eggs often ranges down to 50-60% of the female's fecundity.

Table 19: Parameter values for fecundity and egg viability

Parameter	Definition and units	Basis	Values
trout-spawn-fecund-mult	Multiplier in fecundity equation (eggs per cm spawner length)	Van Winkle et al (1996), Meyer et al. (2003)	Brown Trout: 0.11 Cutthroat Trout: 0.18
trout-spawn-fecund-exp	Exponent in fecundity equation (unitless)	Van Winkle et al (1996), Meyer et al. (2003)	Brown Trout: 2.54 Cutthroat Trout: 2.51
trout-spawn-egg-viability	Fraction of fecundity that become viable eggs in redd (unitless)	Estimate	0.8

The third spawning step is to represent superimposition, the potential mortality of eggs in other redds in the same cell. Superimposition occurs when one spawner digging a redd displaces eggs from an existing redd, making those eggs vulnerable to being eaten or deposited where they cannot survive. In heavily used spawning gravels, superimposition can be high and even limit the reproductive capacity of a stream (McNeil 1964). Previous versions of inSTREAM represented superimposition as part of the redd survival schedule (Sect. 5), but we now represent it as a one-time action of a spawner; this change has no effect on simulated rates of superimposition. The spawner simply causes all existing redds in the spawning cell to execute the superimposition mortality submodel described in Sect. 9.36. InSTREAM's formulation neglects redd guarding by females: more than one redd can be created in the same cell on the same day, with superimposition. It also neglects variation among individual spawners in the size and depth of redds.

In the fourth step, a spawner uses the mate selection submodel of Sect. 9.30 to identify a male that also spawns.

In the final spawning step, the spawner and her selected mate incur a weight loss that represents the energetic cost of spawning. Hayes et al. (2000) cite literature indicating a typical loss, for both male and female trout, of 18% of mass and 45% of energy content upon spawning (e.g., Lien 1978). This energy loss can significantly affect the habitat selection and survival of spawners, so it is included in inSTREAM. When any trout—male or female—spawns, its weight is reduced according to the parameter ***trout-spawn-wt-loss-fraction***: its value of *trout-weight* is multiplied by (1.0 - ***trout-spawn-wt-loss-fraction***) and its value of *trout-condition* recalculated for the new weight. Hayes et al. (2000) support a value of 0.2 for ***trout-spawn-wt-loss-fraction***. This loss affects habitat selection in future time steps, causing spawners to take additional risks to regain weight and reduce starvation risk. Under typical conditions, a 20% loss of weight and condition reduces the probability of survival for 90 days by about 10-15%, while a 30% loss reduces this probability by about 40% (Railsback et al. 2009).

9.29 Spawning site selection

This submodel defines how female spawners select the cell in which they build a redd[56]. While selection of habitat for foraging is modeled mechanistically (Sect. 9.13), selection of spawning habitat is modeled in a simple, empirical way, with spawning cells chosen using preferences for depth, velocity, and substrate observed in real trout. We chose this design because a detailed, mechanistic representation of spawning habitat selection would require considerable additional complexity: it would require modeling processes such as intergravel flow and water quality, which are extremely data-intensive and uncertain. This additional complexity is not necessary to meet inSTREAM's objectives (Sect. 3.1). We do, however, need a simple representation of how flow affects redd placement because a redd's location affects its survival of dewatering (Sect. 9.34) and the habitat conditions that its newly emerged juveniles are exposed to.

Spawning site selection is driven in part by the cell variable *cell-frac-spawn* (Sect. 4.1.3). This variable is input to the model (Sect. 9.1), so it must be evaluated from field observations (Sect. 22.2 provides guidance). The variable represents the estimated fraction of cell area made up of spawning gravel usable by the species being modeled.

We assume that spawning habitat selection is driven only by cell depth, velocity, and spawning gravel availability. Selection is not affected by the size of spawning gravel patches (the cell area with spawning gravel), nor by whether any redds already exist in a cell (discussed in Sect. 9.36). We also ignore variation in spawning gravel quality among cells. These simplifying assumptions could be replaced when necessary to address specific spawning issues.

The first spawning site selection step is for the spawner to identify all the cells that are potential spawning sites. This step uses exactly the same method used by model trout to identify potential destination cells during habitat selection (Sect. 9.13.1): spawners are assumed to select a cell within a radius that defines the area of habitat they are familiar with, using the parameters ***trout-move-dist-A*** and ***trout-move-dist-B***. This formulation does not cause, or allow, long spawning migrations.

For simulations with multiple habitat reaches, the potential spawning cells could include cells in a different reach from the spawner's current cell. That means that a spawner could place its redd in a cell in another reach even if the habitat criteria for spawning (Sect. 9.27) are not all met in that other reach. This possibility occurs because we judged it not important enough to justify the complexity of preventing it.

After a spawner identifies potential spawning cells, it excludes from consideration those with area of spawning gravel (*cell-area* × *cell-frac-spawn*) less than the trout parameter ***redd-area***. However, if no cells meet this gravel-area threshold then no cells are excluded: we assume that if no cells offer sufficient gravel area the trout still spawns and selects a cell by considering only depth and velocity. The parameter ***redd-area*** is an estimate of the area (cm^2) of a typical redd; its meaning and values are discussed in Sect. 9.36.

The second step is for the spawner to rate the potential spawning cells by calculating the variable *spawn-suitability* for each, where:

$spawnSuitability = depthSuitability \times velocitySuitability$.

[56] Selection of the spawning cell is coded in the reporter `spawn-cell`.

The variables *depth-suitability* and *velocity-suitability* are unitless habitat suitability factors determined using methods described below.

The third step is selecting a cell. If no cells offer *spawn-suitability* greater than zero, then the spawner places its redd in its current cell instead of using the method in the following paragraph. (This condition will very rarely occur in most applications, especially if the value of ***reach-max-spawn-flow*** is well-chosen.)

The spawner does not simply select the cell with highest value of *spawn-suitability* (as in previous versions of inSTREAM), but instead selects one of the cells with *spawn-suitability* close to the highest value. This assumption prevents the unrealistically high superimposition redd mortality (Sect. 9.36) that can result from assuming trout precisely identify the most-suitable cell. The parameter ***trout-spawn-suitability-tol*** represents uncertainty or tolerance in a spawner's rating of spawning suitability: the spawning cell is selected from "suitable cells", which are the potential spawning cells with *spawn-suitability* greater than (1.0 - ***trout-spawn-suitability-tol***) × S_{max} where S_{max} is the highest value of *spawn-suitability* among all the potential spawning cells. Users can explore how the value of ***trout-spawn-suitability-tol*** affects superimposition mortality of redds; we preliminarily suggest a value of 0.1, so that cells with suitability greater than 90% of S_{max} are considered for spawning.

Selection of the spawning cell from the suitable cells uses a weighted random draw ("roulette-wheel selection"), in which the probability of a cell being chosen is proportional to its area of spawning gravel. (The random draw method is fully described in the NetLogo user manual section on the "Rnd" extension.) This method makes spawners choose a cell that has gravel while giving each unit area of spawning gravel equal probability of being chosen. (For example, a cell with 2000 cm^2 of gravel has twice the probability of being chosen as a cell with 1000 cm^2.) Spawners will not choose a cell with no gravel unless none of the suitable cells has any gravel; in that case, all suitable cells have equal probability of being chosen.

The suitability factors *depth-suitability* and *velocity-suitability* represent the tendency of salmonids to select fairly well-defined ranges of depth and velocity for spawning (e.g., Knapp and Preisler 1999; Gortázar et al. 2012). Presumably, real trout select these ranges because they correspond to hydraulic conditions favorable to egg survival. For example, intermediate depths have highest suitability probably because redds placed in shallow water are susceptible to dewatering if flows decline and redds in deep water are more vulnerable to scouring during high flows or siltation during low flows. Intermediate velocities have highest suitability, presumably because low velocities provide inadequate flow of water through the redd (important for providing oxygen and removing wastes) and high velocities present a risk of scouring. Depth and velocity suitability functions are certainly a simplification of how salmonids select spawning habitat, but they are an appropriate simplification for inSTREAM and the literature provides them for a variety of species and sites (e.g., Gard 1997; Gortázar et al. 2012; Louhi et al. 2008).

The values of *depth-suitability* and *velocity-suitability* are interpolated linearly from suitability relations provided as parameters. These parameters are referred to as ***trout-depth-suitability*** and ***trout-velocity-suitability*** although each of these parameters is actually a table of (a) depths or velocities and (b) corresponding suitability values. The suitability values normally range between 0.0 and 1.0 but are allowed to have any positive range, e.g., to give velocity stronger effect than depth. The parameter tables (e.g., Table 20) can have any number of rows (X-Y pairs). Figure 39 illustrates example values estimated from several Brown Trout spawning criteria sets (PG&E 1994).

Table 20: Example values for ***trout-depth-suitability*** and ***trout-velocity-suitability***.

Cell depth (cm)	***depth-suitability***	Cell velocity (cm/s)	***velocity-suitability***
0	0.0	0	0.0
5	0.0	10	0.0
50	1.0	20	1.0
100	1.0	75	1.0
1000	0.0	100	0.0
		1000	0.0

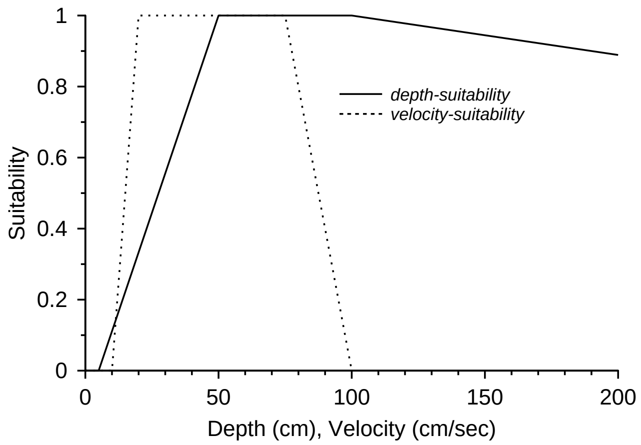


Figure 39: Spawning depth and velocity suitability relations, for the parameter values in Table 20.

9.30 Spawning mate selection

When a female spawns, it attempts to select a male that also spawns[57]. The only purpose of identifying a male spawner is to impose spawning weight loss (Sect. 9.28) on the male. The female selects as its mate the largest fish in the simulation that meets all the male spawner criteria listed below. The largest male is chosen because larger males are assumed more likely to be sexually mature (Meyer et al. 2003), and more likely to compete successfully to fertilize females (Fleming and Reynolds 2004; Jones and Hutchings 2002).

If several females spawn on the same day, the male selected by each female becomes unavailable for subsequent females. If no male meets the criteria as a spawner, there is no effect on the female: it still produces a fertile redd and incurs weight loss due to spawning. This assumption is made because spawning failure due to absence of males is considered too rare and unpredictable

[57] Mate selection is implemented in the procedure spawn.

to include in the model (e.g., Jonsson and Jonsson 2011 indicate that males typically remain in spawning areas longer than females do and can spawn multiple times). Males do not move as a result of spawning because such movement is assumed to have negligible effect.

To identify a male spawner (if there is one), a spawning female identifies the largest male trout that:

Is of the same species as the female;

Occupies the same reach as the female (after the female has moved to her spawning cell);

Has length greater than ***trout-spawn-min-length***;

Has age equal to or greater than ***trout-spawn-min-age***;

Has condition greater than ***trout-spawn-min-cond***; and

Has not previously spawned during the current spawning season, as defined in Sect. 9.27.2.

9.31 Redd survival

The redd survival submodel determines how many of a redd's eggs survive or die; it is executed for each redd on each time step[58]. (This submodel does not include superimposition mortality of eggs, which inSTREAM 7 treats as part of spawning; Sect. 9.28.) For each specific source of redd mortality (except scouring, explained below), the redd survival model uses the relevant submodel (described in the following sects. 9.32 to 9.35) to calculate the mean rate of egg mortality for the time step and then stochastically determines how many eggs die. The following steps are used for each mortality source.

First, the mean mortality rate for the time step is calculated. The daily survival rate (expected fraction of eggs surviving for one day, given the current habitat conditions) is calculated by the relevant mortality submodel, then converted to a mortality rate (here, *M*) by subtracting it from one. Second, the number of eggs that die is then drawn from a random Poisson distribution with the distribution parameter set to *M* × *redd-num-eggs* × *step-length*. (This distribution models the integer number of times that an event occurs—here, an egg dies—within a time period, given the mean rate of occurrence for a unit time— *M* × *redd-num-eggs*—multiplied by the length of the time period (*step-length* days).) This Poisson distribution draw can produce a number greater than *redd-num-eggs*; in that case, the number of eggs that die is set to *redd-num-eggs*. (Previous versions of inSTREAM modeled egg mortality using a random Bernoulli distribution, which models the number of times some event happens in a specified number of trials, given the probability of one event per trial. Using the Poisson distribution is expected to have negligible difference and allows use of NetLogo's efficient code for it.)

Scouring mortality is an exception to the above steps because it is modeled as Boolean: a redd's eggs either all survive or all die in a scouring event (Sect. 9.35).

Finally, the number of eggs that die are subtracted from the redd's variable *redd-num-eggs*.

The model executes the redd mortality submodels sequentially, so eggs killed by one mortality source are subtracted from the number still alive before the next is evaluated. The order in which

[58] Redd survival is programmed in procedure `redd-survive`.

redd mortality sources are executed is defined in Sect. 5. If *redd-num-eggs* reaches zero, the empty redd is immediately removed from the simulation.

We selected and designed the redd mortality submodels to reflect inSTREAM's primary purpose of predicting how flow and temperature regimes affect trout populations. For example, spawning gravel quality has several effects on redd success (Kondolf 2000) but is not represented in inSTREAM because those effects are at most indirectly related to flow regime. For some redd mortality sources (especially, dewatering), more detailed and mechanistic approaches are available in the literature and could be added to inSTREAM in situations where these mortality sources are believed to be important.

Users may encounter an artifact of the redd mortality methods that can make simulated egg survival slightly higher than expected when survival probabilities are low. The combination of an integer (Poisson) distribution for the number of eggs that die and inSTREAM 7's short time steps yield many time steps with zero egg deaths even when daily survival probability is relatively low. InSTREAM 7 can produce four time steps of zero mortality per day when a single daily time step would produce several deaths. This artifact is unlikely to significantly affect inSTREAM population-level results but may be apparent in tests of egg survival results.

9.32 Low temperature redd mortality

Both low and high temperatures cause mortality in eggs, over different ranges from those causing mortality in fish. Mortality due to high and low temperatures are modeled separately. Logistic functions represent the available data well[59].

The daily fraction of eggs surviving low temperatures is modeled as an increasing logistic function of temperature. Parameter values appear to differ among trout species, with differences especially likely between species (or stocks) that spawn in the fall versus spring. Developing parameter values from published data on egg survival typically requires careful conversion of the time scale: published data is often reported as survival over the duration of an experiment, which much be converted to the daily scale used for inSTREAM's parameters. For example, 80% survival over a 60-day incubation period is equivalent to daily survival probability of 0.9963 ($0.9963^{60} = 0.8$). Because eggs incubate slowly at low temperatures, even apparently high daily survival rates can result in low egg survival over the entire incubation period.

Parameter values for spring-spawning Rainbow Trout and fall-spawning Brown Trout (Table 21; Figure 40) were estimated from data compiled by Brown (1974). For Rainbow Trout, Brown (1974) indicates that eggs have a 90 percent survival rate over a 100-d incubation period at 3° C (daily egg survival = 0.999). We assumed a daily survival rate of 0.9 (very low long-term survival) for 0° C, and fit logistic parameters to these survival rates at 0 and 3° C. For Brown Trout, Brown (1974) cited data indicating that Brown Trout egg incubation can take over 150 days at very low temperatures. Parameter values for Brown Trout were estimated by assuming 90 percent egg survival over 150 days at 1º C (daily survival of 0.9993) and daily survival of 0.9 at 0º; however, Ojanguren and Braña (2003) and Réalis-Doyelle et al. (2016) provide and cite data on how Brown Trout egg survival varies with temperature.

[59] Calculation of low temperature survival is in the reporter `redd-survival-lo-temp-for`.

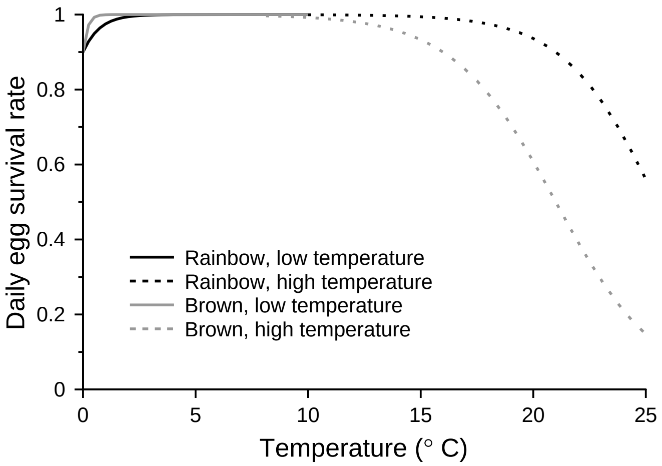


Figure 40: Redd survival rates for low and high temperature, using example parameters of Table 21.

Table 21: Parameters and values for low and high temperature mortality in redds

Parameter	Meaning and units	Species, parameter value source	Values
mort-redd-lo-temp-T1	Temperature (° C) at which the mean daily low temperature survival rate is 0.1	Rainbow Trout; analysis of data compiled by Brown (1974)	-3.0
		Brown Trout; analysis of data compiled by Brown (1974)	-0.8
mort-redd-lo-temp-T9	Temperature (° C) at which the mean daily low temperature survival rate is 0.9	Rainbow Trout; analysis of data compiled by Brown (1974)	0.0
		Brown Trout; analysis of data compiled by Brown (1974)	0.0
mort-redd-hi-temp-T1	Temperature (° C) at which the mean daily high temperature survival rate is 0.1	Rainbow Trout; analysis of data by C. Myrick (Department of Fish, Wildlife, and Conservation Ecology, University of California, Davis, pers. comm. with S. Railsback, 1998)	30
		Brown Trout; estimate	26
mort-redd-hi-temp-T9	Temperature (° C) at which the mean daily high temperature survival rate is 0.9	Rainbow Trout; analysis of data by Myrick	21
		Brown Trout; estimate	16

9.33 High temperature redd mortality

High temperatures can be directly lethal to trout eggs, while sub-lethal temperatures can promote fungus and disease[60]. The fraction of eggs surviving high temperatures is modeled as a decreasing logistic function of temperature (Figure 40). Parameter values for Rainbow Trout (also used for Cutthroat Trout by Railsback and Harvey 2002) are based on interim results of lab studies conducted by Myrick (1998). These data showed daily survival rates declining from about 0.9998 at 11° C to about 0.985 at 19° C. The resulting parameter values (Table 21) appear to indicate high survival at high temperatures, but in fact cause low survival if temperatures remain elevated for long periods. Fall-spawning trout are likely to be less adapted to high

[60] Calculation of high temperature survival is in the reporter `redd-survival-hi-temp-for`.

incubation temperatures. We arbitrarily set the parameter values for Brown Trout in Table 21 to 5º less than the Rainbow Trout values, but Ojanguren and Braña (2003) and Réalis-Doyelle et al. (2016) provide and cite relevant data. (Martin et al. 2017 discuss and model high temperature mortality in salmonid eggs and suggest a mechanism that may cause temperature to have stronger effects at lower temperatures than indicated by laboratory studies: limited oxygen transfer at low water velocities.)

9.34 Dewatering redd mortality

Dewatering mortality occurs when declining flow exposes a redd to air; eggs can be killed by desiccation or the buildup of waste products that are no longer flushed away[61]. Reiser and White (1983) did not observe significant mortality of eggs when water levels were reduced to 10 cm below the egg pocket for several weeks. However, they also cited literature indicating high mortality when eggs and alevins are only slightly submerged (which may yield poorer chemical conditions than being dewatered), and high mortality for dewatered alevins. Because inSTREAM does not distinguish between eggs and alevins, these processes are not modeled mechanistically or in detail. The dewatering survival function is simply that if depth is zero then the daily fraction of eggs surviving equals the fish parameter ***mort-redd-dewater-surv***. This parameter has a suggested value of 0.9, which reflects the variability in dewatering effects. Egg survival may be high when a redd is first dewatered, which provides further support for ***mort-redd-dewater-surv*** values near 1.0.

9.35 Scour redd mortality

This submodel represents redd mortality due to both scour and deposition, which occur when high flows move streambed sediment, eroding some areas and depositing sediment on others[62]. Eggs scoured out of a redd face very high risk of mortality. Deposition of new sediment on top of a redd can reduce water flow through the redd so that inadequate transport of oxygen and waste materials kills eggs, and may prevent trout from emerging. Deposition is especially likely to reduce survival if it includes fine sediment. Redd scour mortality can be very important to trout populations and communities. For example, Strange et al. (1992) found that the frequency of winter redd scour events determined the relative abundance of brown vs. Rainbow Trout in a Sierra Nevada stream.

The mechanisms of scour in salmonid spawning gravel have been studied in detail by Wilcock et al. (1996), Haschenburger (1999), May et al. (2009), Cienciala and Hassan (2013), and Gauthey et al. (2017). These field studies all support several conclusions about modeling redd scour. First, they support the use of Shields stress at the peak flow as a predictor of scour during high flows. (Shields stress is a dimensionless measure of the force exerted by flow on a streambed, commonly used in modeling sediment transport.) For example, May et al. (2009) found a sharp threshold in Shields stress above which gravel movement became widespread, and several of these studies found strong, nonlinear relations between reach-averaged Shields stress and the extent of gravel disruption. Second, the relation between local (e.g., at the scale of cells in inSTREAM) Shields stress and gravel movement is highly stochastic: while the relation between Shields stress and scour can be strong at a reach scale, whether gravel at any particular location

[61] Calculation of dewatering survival is in the reporter `redd-survival-dewater-for`.

[62] Redd scour survival is coded in the procedure `redd-survival-scour-for`. Whether flow has peaked and the reach shear stress parameter are calculated in the procedure `update-redd-scour-vars`, which is called from `update-habitat`.

moves is much less predictable. Factors contributing to unpredictability include hydraulic features at scales finer than cells and that scour in one event can depend on how gravel moved in previous events. A third conclusion is that the vulnerability of a redd to scour depends strongly on the redd depth: shallow redds are much more vulnerable than deep ones.

Given the difficulty and uncertainty of predicting scour at the cell scale, we choose instead to use the reach-scale model of Haschenburger (1999) and then apply its estimate of scour probability to each redd independently. This approach was developed for gravel-bed channels and may not be well-suited for sites where spawning gravels occur mainly in pockets behind obstructions (where scour is likely even less predictable).

Haschenburger (1999) observed the spatial distributions and depths of scour and deposition that resulted from a number of high-flow events in several study sites in gravel-bed rivers. She found the proportion of a stream reach scoured or filled to a specified depth during an event to follow an exponential distribution, the parameter for which (*scour-param*) varied with site-average dimensionless Shields shear stress. Therefore, inSTREAM assumes that the probability of a redd being destroyed equals the proportion of the stream reach scouring or filling to depths greater than the value of the redd parameter ***mort-redd-scour-depth*** (cm). Consequently, the probability of a redd not being destroyed (*scour-survival*) equals the proportion of the stream scouring or filling to a depth less than the value of ***mort-redd-scour-depth***. This scour survival probability is estimated from the exponential distribution model of Haschenburger (1999); the proportion of the stream scouring to less than a given depth is the integral of the exponential distribution between zero and the depth:

$$scourSurvival = 1.0 - exp(-1 \times scourParam \times mortReddScourDepth).$$

(The value of *scour-survival* is set to 1.0 if *scour-param* × ***mort-redd-scour-depth*** exceeds 100, to prevent the exponential function from producing a variable underflow.)

Haschenburger modeled *scour-param* empirically as:

$$scourParam = 3.33 \times \exp\big(-1.52 \times (shearStress/0.045)\big)$$

where *shear-stress* is the peak Shields stress (measured at the reach scale) occurring during the high-flow event. Shields stress increases with flow, a relationship represented in inSTREAM (as in many sediment transport models) as a power function of flow:

$$shearStress = reachShearA \times flow^{reachShearB}$$

where ***reach-shear-A*** (s/m^3) and ***reach-shear-B*** (unitless) are reach parameters with highly reach-specific values. Methods for estimating ***reach-shear-A*** and ***reach-shear-B*** are discussed in Sect. 23.1. Setting ***reach-shear-A*** to 0.0 essentially turns off redd scour mortality by making it extremely unlikely at all flows.

The fish parameter ***mort-redd-scour-depth*** can be evaluated as the egg burial depth, the distance down from the gravel surface to the top of a redd's egg pocket. Scour to this depth almost certainly flushes eggs out of the redd. Deposition of new material to this distance would double the egg pocket's depth and make egg survival and emergence much less likely. DeVries (1997) reviews egg burial depths for stream trout. Values of 5-10 cm are reasonable for small trout using relatively small gravel. Zimmer and Power (2006) and Gauthey et al. (2017) observed redd depths of 5-15 cm in small Brown Trout.

The scour mortality submodel uses the following algorithm for each redd.

First, the submodel determines whether the current time step is the peak of a high-flow event. A flow peak occurs if (a) the current value of flow is the highest of all the flows in the input for times between 2.0 d before and 2.0 d after the current value of *sim-time*, and (b) the value of *flow* has changed since the previous time step. Note that:

The set of input flows associated with times ± 2.0 d of *sim-time* are not necessarily the values of *flow* for the time steps within two days (Sect. 8.1).

The 2.0 d time span used to identify a peak is not a parameter but is easily changed in the model code.

This approach can produce flow peaks during periods of relatively stable, moderate flows, but under such conditions the probability of redd scour will be near zero.

The check that a peak flow cannot equal the flow in the previous time step prevents redd scour from being executed repeatedly in consecutive time steps when flow input uses a longer time scale than model time steps. For example, daily flow input results in all time steps of each day having the same value of flow; but redd scour should be executed only once per value of flow input, not on each time step of a day.

If the current time step is not at a flow peak, then no further steps are executed and there is no scour mortality. If it is at a peak, the above equations are used to calculate *scour-survival*. Then scour survival is evaluated as a Bernoulli trial with *scour-survival* as its parameter. If the trial produces a value of false, the redd is assumed scoured and all its eggs die; otherwise, all eggs survive.

(An alternative version of inSTREAM 7 models shear stress at the cell scale. It uses cell-level relations between flow and shear stress imported from a hydraulic model, just as the relations between flow and depth and velocity are. Then the above equation for *scour-survival* is applied to each cell. This version is available from the authors upon request.)

9.36 Superimposition redd mortality

Unlike the other redd mortality submodels, superimposition is not executed repeatedly as part of the redd survival schedule. Instead, a redd executes it only when a new redd is created in the same cell. The spawner creating the new redd causes each existing redd to execute this submodel (Sect. 9.28)[63]. Because superimposition is not part of the redd survival submodel, this submodel determines not the egg survival rate but the number of eggs that die.

Superimposition typically causes mortality of many but not all eggs in a redd (Essington et al. 2000, Hendry et al. 2003). For simplicity, inSTREAM assumes that superimposition is accidental with no bias for or against spawning over existing redds. Essington et al. (1998) and Gortázar et al. (2012) provide evidence that stream trout may intentionally superimpose their redds over existing ones, a practice that has the advantages of reducing (a) the work necessary to clean redd gravels and (b) the competition that the spawner’s offspring will face (Morbey and Ydenberg 2003). The formulation could be modified to represent intentional superimposition and the complex effects it might have. Such a formulation could be supported by site-specific

[63] Superimposition is programmed in the procedure `be-superimposed-by`.

observations, but reasonable general assumptions about intentional superimposition remain elusive.

Superimposition redd mortality is modeled as a function of the area disturbed in creating the new redd and the area of spawning gravel available. A redd executes the following steps each time a new redd is created in its cell.

First, the probability P_{su} of the new redd causing superimposition is calculated as the area of the new redd divided by the area of spawning gravel in the cell. The new redd's area is the value of the trout parameter ***redd-area*** for the species of the new redd. The area of spawning gravel in the cell is simply equal to *cell-area* × *cell-frac-spawn*. If P_{su} is greater than 1.0, it is set to 1.0. However, P_{su} is set to 0.0 if *cell-frac-spawn* is zero; we assume superimposition is impossible in the absence of spawning gravel.

The probability P_{su} is then used in a random Bernoulli trial to determine whether superimposition occurs. If the trial produces a result of true, the number of eggs that die is drawn from a uniform integer distribution with range of possible values from zero to *redd-num-eggs*.

Because of how the parameter ***redd-area*** is used in this formulation, it is defined as the area a spawner excavates in creating a new redd. Our field observations at the Little Jones Creek site suggest a ***redd-area*** value of 1200 cm^2 (the area of a circle with a diameter of 35 cm) for relatively small trout. Zimmer and Power (2006) report values around 10,000 cm^2 for Brown Trout in the range of 26-36 cm.

9.37 Redd development

The redd development submodel determines when the eggs in a redd begin to emerge as free-swimming juveniles[64]. InSTREAM intentionally ignores the developmental stages between eggs and free-swimming juveniles. Development is tracked by the variable *redd-frac-developed*, which is initialized to 0.0 when redds are created, and eggs start turning into fish when *redd-frac-developed* reaches 1.0. This submodel determines the daily increase in *redd-frac-developed*.

Temperature has long been recognized as the primary driver of salmonid egg development (e.g., Embody 1934). Other factors such as dissolved oxygen concentration (as affected by water quality or siltation) and disease can affect development but are not considered essential for the purposes of inSTREAM. Published models of development rate often quantify degree-days to emergence, although a variety of observations indicate that development time is not a constant function of degree-days during incubation (e.g., Réalis-Doyelle et al. 2016; Wood and Fraser 2015).

Most of the literature relating salmonid egg development to temperature relies on experiments that exposed eggs to a constant temperature, but temperature is not constant in inSTREAM. We therefore adapt the approach of Van Winkle et al. (1996) of modeling the increase in *redd-frac-developed* each time step as a quadratic function of temperature during that step. On each time step, the value of *redd-frac-developed* is increased by a development increment *D*, where:

$$D = stepLength \times \{reddDevelA + (reddDevelB \times T) + (reddDevelC \times T^2)\}.$$

The values of the three redd parameters ***redd-devel-A***, ***redd-devel-B***, and ***redd-devel-C*** can be estimated from published relations between temperature and incubation time. For example,

[64] Redd development is coded in the procedure `develop-eggs`.

results of a typical study that measures incubation time (d) at several temperatures can be used by assuming that development increment *D* is equal to 1.0 divided by observed incubation time. The parameter values from Réalis-Doyelle et al. (2016) in Table 22 were estimated in this way, using Excel's Solver to find the parameter values that best fit the data. Réalis-Doyelle et al. (2016) published mean time to first feeding for Brown Trout incubated at five temperatures from 4 to 12°C. The difference between model-predicted and observed incubation times ranged from 1 day (at 4° and 12°) to 8 days at 8°. The two parameter sets for Brown Trout in Table 22 produce substantially different development times, and Réalis-Doyelle et al. (2016) cite other studies of incubation time in Brown Trout, some of which produced results quite different from theirs. It appears that users may need to compare several studies and determine which produce results most compatible with evidence from their application sites.

In estimating redd development parameters, it is important to use data compatible with inSTREAM's definition of incubation time. This submodel represents the time from egg fertilization until new trout are ready to swim and feed, an endpoint typically referred to as "fry swim-up" or the time of first feeding. Some studies report other measures of incubation time such as the time to hatching or yolk sac absorption, which can be substantially shorter than the incubation time we model.

Negative values of ***redd-devel-A*** will produce negative development rates at temperatures near 0°. We recommend avoidance of such values unless temperatures low enough to cause negative development rates will not be simulated commonly.

Table 22 provides parameter values for several species. Parameter values of Van Winkle et al. (1996) for Rainbow Trout, used in previous versions of inSTREAM, had a negative value of ***redd-devel-A*** and therefore produced negative rates at very low temperatures. The values in Table 22 produce the same development rates as those of Van Winkle et al. (1996) at temperatures > 3°, and positive rates at all positive temperatures.

Railsback and Harvey (2001) found the Rainbow Trout parameters in this table reasonable for a Cutthroat Trout population in coastal California.

Table 22: Parameter values for redd development

Species	**Source**	***redd-devel-A*** (unitless)	***redd-devel-B*** (°C^{-1})	***redd-devel-C*** (°C^{-2})
Rainbow Trout (spring spawning)	Van Winkle et al. (1996)	0.0	0.00126	0.0000372
Brown Trout (fall spawning)	Van Winkle et al. (1996)	0.00313	0.0000307	0.0000934
Brown Trout (fall spawning, European)	Réalis-Doyelle et al. (2016)	0.00165	0.00166	0.0000275

9.38 Redd emergence

This submodel represents the creation of new trout from redds; trout biologists refer to the "emergence" of new swim-up fry from redds. The model has two components: emergence timing and creation of new trout. This submodel is executed once per day (not each time step) by each redd that has a value of *redd-frac-developed* of 1.0 or higher[65].

The emergence timing element determines when each new trout is created. Emergence begins on the day when *redd-frac-developed* reaches 1.0, then the new fish emerge over a period of several days. Causing emergence to occur over several days reproduces observed natural variation in emergence timing and can potentially have strong effects on survival of newly emerged trout. These fish compete with each other for food as soon as they emerge. If all emerged on the same day, without time for some to disperse away from the redd, competition would probably be overestimated.

InSTREAM 7 controls the rate at which new trout emerge from a redd in two ways. First, this submodel is executed only on the first time step of the daytime light phase; this keeps the rate of emergence independent of the number or length of time steps. Second, as a simple way to spread emergence over several days, inSTREAM assumes that 10% of a redd's eggs (rounded up to the next integer) emerge on the first day of emergence; 20% of the redd's remaining eggs emerge on the next day; 30% of the remaining eggs emerge on the third day; etc., until 100% of remaining eggs emerge on the 10th day. For example, if a redd contains 105 eggs on the day that development is complete, the number of new trout emerging N_E that day is 11, with 94 eggs remaining in the redd. On the next day (assuming no egg mortality occurs), N_E is 19 (20% of 94, rounded up) with 75 eggs (94-19) remaining. On the third day of emergence, N_E is 23 (30% of 75, rounded up).

This algorithm is implemented using the redd variable *redd-emerge-days* (d), which records the number of days the redd has been producing new trout. The value of *redd-emerge-days* is initialized to 0 when the redd is created and incremented by 1 on each day when it creates new trout.

As emergence proceeds, the eggs remaining in a redd remain susceptible to redd mortality. On the day when the number of remaining eggs reaches zero (typically, when *redd-emerge-days* is 10), the final step in this submodel is to remove the redd from the model.

One new trout is created for each egg that emerges, unless new trout are represented as superindividuals. Superindividuals are created if the value of trout parameter ***trout-superind-max-rep*** exceeds 1. In that case, the number of new trout created is N_E divided by ***trout-superind-max-rep***, rounded up to the next integer.

The new trout have their state variables initialized using these methods:

trout-species is set to the species of the redd creating the trout.

trout-sex is assigned as "male" or "female" randomly, with equal probability. (Morán et al. 2016 show that this assumption is surprisingly uncertain: sex ratios of newly emerged trout can be highly variable.)

trout-age is set to 0 years.

[65] This submodel is implemented in the procedure `redd-emergence`. Initialization of new trout is by the procedure `initialize-trout-with`.

trout-length is drawn from a random triangular distribution specified by the trout parameters ***trout-emerge-length-min***, ***trout-emerge-length-mode***, and ***trout-emerge-length-max***.

trout-condition is set to 1.0 and *trout-weight* is set to the weight of a fish with length of *trout-length* and condition 1.0, using the method in Sect. 9.20.

trout-cell is set to the redd's cell.

trout-superind-rep is set to ***trout-superind-max-rep***, except for the last trout created, which has *trout-superind-rep* set to the remainder of N_E / ***trout-superind-max-rep***. This adjustment keeps the total number of trout represented by the superindividuals equal to N_E.

Other trout variables are initialized as they are for trout created at the start of a simulation (Sect. 7.3).

InSTREAM imposes variation among individuals in length at emergence because habitat selection (and, consequently, growth and survival) is modeled using a length-based hierarchy. Elliott (1994) found only slight length variation among trout emerging from the same redd, but that variation gave larger fish an advantage in dominance likely to persist and grow over time because competition among newly emerged trout can be intense (and this intense competition occurs in inSTREAM; Railsback et al. 2002). InSTREAM neglects any effect of spawner size on initial offspring size; such effects have been observed in some salmonid populations and may be important for modeling problems that include evolution (e.g., Ayllón et al. 2016, 2019a, b). However, spawner effects on offspring size do not appear important for typical applications of inSTREAM.

The parameters for the length distribution of newly emerged trout can be estimated from the literature or hatchery data. June (1981) measured lengths of newly emerged coastal Cutthroat Trout in Washington found in a downstream trap. A few of these fry had lengths between 2.4 and 2.7 cm, but most were between 2.7 and 3.0 cm. Elliott (1994) observed length at emergence for Brown Trout at several sites, producing a mean length of 2.8 cm with a standard deviation of 0.2 cm. Therefore, values of 2.6, 2.8, and 3.0 cm seem reasonable for ***trout-emerge-length-min***, ***trout-emerge-length-mode, and*** ***trout-emerge-length-max***.

9.39 Superindividual separation

This submodel turns a superindividual into individual trout. A superindividual executes it when the growth submodel causes the superindividual to reach a specified length (Sect. 9.20)[66].

The number of new trout created is the superindividual's value of *trout-superind-rep*. For each new trout, all state variables are set to the values of the superindividual, except for *trout-superind-rep,* which is set to 1. The superindividual is then removed from the model.

(The individual trout created from a superindividual are all assigned the same length; their lengths subsequently diverge as a consequence of their luck in the sorting process that determines the order in which trout execute their habitat and activity selection action each time step; Sect. 5.)

[66] The superindividual separation submodel is coded in the procedure `separate-superindividual`.

Part III: Software Guide

10 Introduction to the InSTREAM 7 Software

10.1 Software goals and platform

Since the beginning of the inSTREAM development program, we recognized the importance of software design and implementation for making complex individual-based models (IBMs) usable and credible. We continue to pursue the software goals stated in Sect. 18.4 of Railsback et al. (2009), which are that inSTREAM's software should:

- Allow changes in model formulation to be implemented easily and without a high chance of introducing errors. Changes in the model are inevitable as it is applied to different problems and as we learn more about mechanisms in the model.
- Be thoroughly tested and documented. Undiscovered mistakes are extremely costly, and thorough documentation is essential for the model to become widely used and accepted.
- Promote documentation of changes and version control, so software documentation remains current as the model evolves.
- Provide the graphical and file outputs necessary to understand and develop belief in results.
- Provide tools that make simulation experiments easy to set up and execute.

InSTREAM 7 differs from all previous versions of inSTREAM in being implemented in the NetLogo software platform. Previous versions were implemented using Swarm, which was the best platform available when the inSTREAM program started in 1999. In the meantime, Swarm is no longer maintained and NetLogo (Wilensky 1999) has developed into a powerful and standard platform for IBMs in many fields (Railsback et al. 2017; Railsback and Grimm 2019). The primary advantages of NetLogo are that it:

- Provides built-in structures (grid cells, mobile individuals) and hundreds of powerful programming commands (primitives) specifically for IBMs; these greatly reduce the amount of code needed to implement a model and make it much easier to read and modify code. Many of NetLogo's primitives use sophisticated algorithms to improve execution speed.
- Provides a variety of graphical interfaces and controls.
- Includes a powerful experiment manager (called BehaviorSpace).
- Is professionally maintained and documented, and has a large and active user community.
- Is free, open-source, easy to install, and available for all major operating systems.

10.2 Software license

Like all previous versions, the inSTREAM 7 software is free software distributed under the GNU General Public License as published by the Free Software Foundation. This means that users are free to use and modify the software, but they must make their source code available to anyone they distribute the software to and cannot patent the software or make it proprietary. The full GNU license is distributed with the software and is also available from www.gnu.org/licenses/. The inSTREAM software is copyrighted.

10.3 Testing, maintenance, and evolution

We encourage users to check the inSTREAM web site (identified in Sect. 1.1) for updates as the model and its documentation evolve. Updates and new releases of inSTREAM are likely to result from changes in NetLogo, implementation of new features and correction of mistakes, and adaptation of the model to address new problems.

We test the inSTREAM software thoroughly before each release and make an archive of completed and documented software tests available from the web site. Code testing is normally done by temporarily activating code in the inSTREAM software that provides detailed output from a particular submodel or procedure, and then comparing this output against an independent implementation of the submodel in Excel. We then comment and archive the Excel files to document the tests. About 22 such test files were available for the first release of inSTREAM 7. The code to produce each test output file either remains in the inSTREAM 7 software (commented out) or is in a separate code file available at the inSTREAM 7 web site.

10.4 Overview and getting started

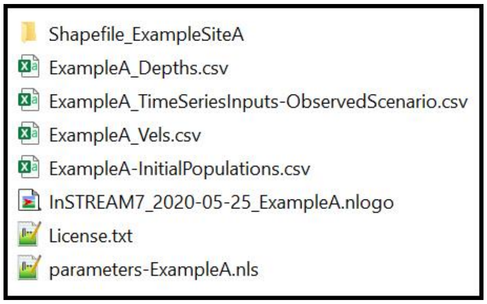
We recommend the following steps for getting started with inSTREAM 7 and applying it to new sites.

1. Install NetLogo, following guidance in Sect. 11.2. Become familiar with the NetLogo platform, especially its graphical interface (Sect. 11.3).
2. Understand the concept of an inSTREAM “project” (Sect. 11.1) and install the inSTREAM 7 software (Sect. 11.2).
3. Try the software using the example project distributed with inSTREAM, learning to control simulations (Sect. 11.3) and understand the output files (Sect. 12).
4. Apply the model to a new site by preparing the input files (Sect. 13) and selecting parameter values to control the simulation and represent site-specific conditions (Sect. 14).
5. Conduct the kinds of simulation analyses discussed in sects. 18-25, using the BehaviorSpace tool (Sect. 15).
6. When problems arise, consult the troubleshooting guide at Sect. 17.

11 Software Installation and Execution

11.1 The project concept

In this document we refer to an inSTREAM 7 “project” as the collection of software and files needed to run the model. While this project concept is not enforced by the software, it is a convenient way to think about and manage the software. A “project” is a directory or directory tree containing all the files needed to run inSTREAM 7 for a particular set of scenarios at a particular site (Figure 41); when the model is run, its output files become part of the project.



Shapefile_ExampleSiteA
ExampleA_Depths.csv
ExampleA_TimeSeriesInputs-ObservedScenario.csv
ExampleA_Vels.csv
ExampleA-InitialPopulations.csv
InSTREAM7_2020-05-25_ExampleA.nlogo
License.txt
parameters-ExampleA.nls

Figure 41. Example inSTREAM 7 project. In this example, all input files except the GIS shapefile are in the main project directory. Alternatively, the other input files (those with .csv extensions) could be in their own subdirectory. The project contains its own customized copy of the inSTREAM 7 software and parameter file.

A project must include the following files; input file formats are explained in Sect. 13.

- The inSTREAM 7 NetLogo file, which includes its code. This file always has a file extension of `.nlogo` and typically is named something like `inSTREAM7.0.2-MyProject.nlogo`. You will very likely make changes to this file for each project, so it is wise to rename it to include its project name. The NetLogo file will always be in the main project directory, not in a subdirectory.
- A parameter file, which is a text file containing the NetLogo code that sets the values of all the model parameters (explained at Sect. 14). This file always has a file extension of `.nls` (the extension for secondary NetLogo code files) and typically is named something like `parameters-MyProject.nls`. Again, you will edit this file to make it specific to your project, so it is smart to give it an identifying file name.
- A GIS "shapefile" in ESRI format, containing cell input. A shapefile is actually a set of several files, so these are normally placed in a subdirectory under the main project directory.
- The time-series input, initial population, and hydraulics input files. These can be placed either in the main project directory or in a subdirectory.
- The file `License.txt` is normally distributed with inSTREAM 7; it contains the software license text and is not needed for model use.

The parameter file provides the names of the shapefile and all input files, including the subdirectory that they are (optionally; Sect. 13.1) in. When the model executes, all output files are written into the same directory as the NetLogo file, the main project directory.

All project files—except output files—are typically small, so we recommend creating new projects for each simulation experiment instead of trying to keep input and results from several simulations in the same directory. Creating a new project simply requires copying an existing project into a new directory and re-naming any files altered for the new project.

11.2 Installing NetLogo and inSTREAM 7

NetLogo must be installed on the computer used to run inSTREAM 7. NetLogo is professionally packaged and readily installed from its download site: http://ccl.northwestern.edu/netlogo/.

Currently, inSTREAM 7 works with version 6.2 (released in late 2020) and will likely be compatible with future NetLogo releases. However, inSTREAM 7 will not work without modification for earlier releases of NetLogo because of changes in how the “time extension”—the code inSTREAM uses to handle dates and times—was packaged. Starting with version 6.2, this extension is included with the standard NetLogo distribution.

Installing inSTREAM 7 is simply a matter of obtaining an existing project directory. We distribute the model from our web site (Sect. 1.1) in the form of a zip-archived example project that looks like Figure 41. Simply download the zip file and extract it.

The troubleshooting guide (Sect. 17) provides help with a number of issues likely to arise during installation and early use of inSTREAM; especially, see its entry about memory availability and increasing the memory available to NetLogo.

11.3 Becoming familiar with NetLogo

Using inSTREAM 7 requires familiarity with NetLogo’s basic concepts and interfaces. NetLogo provides extensive introductory learning materials that we do not attempt to reproduce here. Railsback and Grimm (2019) and Wilensky and Rand (2015) provide more complete introductions to modeling with NetLogo.

We strongly suggest that new users open NetLogo (from the normal Start menu or by double-clicking on the inSTREAM 7 NetLogo file) and use the introductory tutorials and guides available in its User Manual. In particular we recommend:

- From the Help menu, select the User Manual, which opens in a web browser.
- The User Manual menu appears on the left side of the browser window. Select **Tutorial #1: Models** and work through the tutorial.
- At least briefly review the other User Manual content. This content can be used as needed to understand specific parts of the model interface and code.

11.4 Running and observing simulations

In this section we describe the specific tools provided for running and observing inSTREAM 7 simulations. Many of these are illustrated in Figure 42, which shows the user interface that opens when you either double-click on the inSTREAM 7 NetLogo file or start NetLogo and use its File menu to select the inSTREAM 7 NetLogo file.

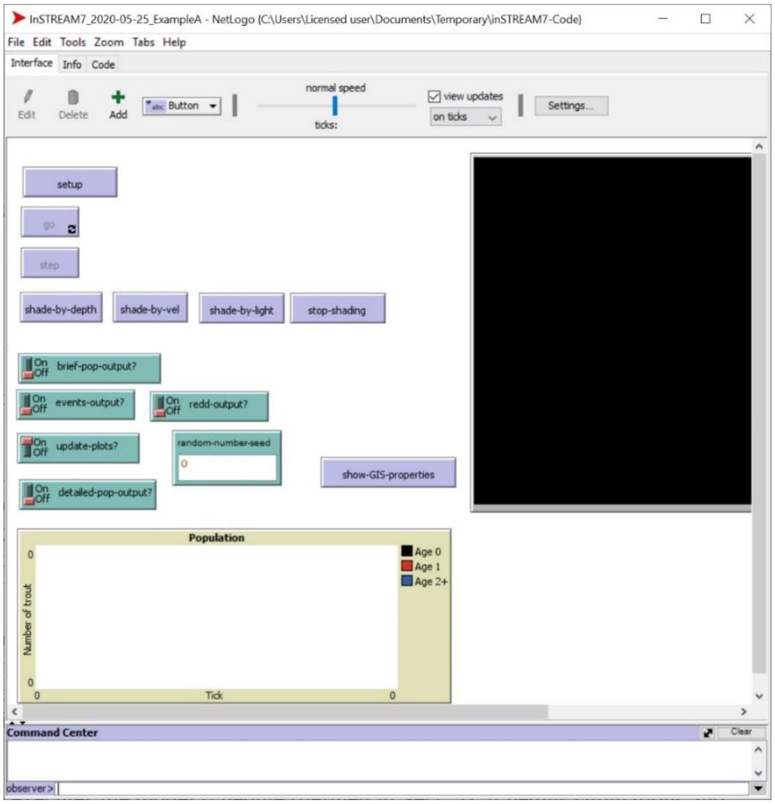


Figure 42. Example inSTREAM 7 interface.

11.4.1 NetLogo’s tabs

NetLogo provides three tabs that you will move among. The **Interface** tab (shown in Figure 42) provides the graphical interface and controls normally used to run and observe the model.

The **Info** tab simply provides text information about the model. This information normally specifies the version of inSTREAM 7 implemented in the NetLogo file, and provides a log of any minor changes made to it. You can add your own notes to the Info tab (click “edit” to toggle its write mode) to keep track of your own changes and differences among projects.

The **Code** tab contains the model code in NetLogo’s own programming language. Normal use does not require working in the Code tab. However, when the code detects an error (e.g., missing or incorrect input) it will display an error message and take you to the Code tab and indicate which code was executing when the error occurred. Normally the error message itself will be sufficient to let you fix the problem without understanding the code, so you can simply click back to the Interface tab.

11.4.2 Key interface buttons

As you learned from the NetLogo tutorial, there are two buttons that appear on the interface of any NetLogo model: **setup** and **go**. Clicking the **setup** button executes the procedure that initializes the model, building the simulated stream habitat and creating the initial trout population.

The **go** button then executes the model schedule (defined in Sect. 5): it begins simulations and continues them until the model's stop date is reached. However, clicking go a second time pauses the simulation at the end of the current time step. The go button can be used as a toggle to stop and re-start execution.

The **step** button also executes the model's schedule, but only once: it lets the user run one time step at a time.

11.4.3 Other buttons

There are typically several additional buttons on the inSTREAM interface. (NetLogo makes it easy to add such controls, so the interface must be expected to vary among and within model versions.) Four other buttons illustrated in Figure 42 control how stream habitat is displayed during executions. Clicking the **shade-by-depth** button tells inSTREAM to color each habitat cell by its depth, and the **shade-by-vel** button shades cells by velocity. (Shading is updated only when the flow changes, which may take several time steps.) The **shade-by-light** button switches shading to show the light intensity that fish experience in each cell. **Stop-shading** turns shading off, which increases execution speed.

The **show-GIS-properties** button simply shows, in the Command Center, the names of all the "properties" of the cell polygons in the GIS shapefile. It also displays a few example values of each property, from randomly selected cells. This button is useful for setting the GIS property name parameters (Sect. 14.1.2).

11.4.4 File switches

The model interface also contains switches that turn different kinds of file output (described in Sect. 12) off and on. Output files can be turned off and on during a simulation, and can even be turned on for the first time partway through a simulation.

11.4.5 Inputs

An "input" is an interface element that lets the user see and change the value of a model parameter. InSTREAM typically has only one input, labeled **random-number-seed**. This input is how the user sets the value of the ***random-number-seed*** parameter explained in Sect. 6.9.

11.4.6 View and inspectors

NetLogo's "View" displays the simulated stream and trout (Figure 43). As the NetLogo tutorial indicates, you can right-click on the View and open "inspectors" to see (and change) the variables of objects in the model. Right-clicking opens a menu that includes options to "inspect" any object near where you clicked. These objects (which, except for patches, are defined in Sect. 4.1) can include:

- A "patch": Patches are NetLogo's built-in square grid cells, which are very small in most inSTREAM 7 applications; there are several patches per cell. Patches are used by inSTREAM only for display and therefore do not appear in the model description. They have variables for habitat characteristics that can be displayed via patch colors: depth, velocity, and light (irradiance); and for which cell and reach the patch belongs to.
- A cell: Cells represent inSTREAM's polygonal habitat cells. The view displays cell boundaries as white lines, but the cell itself is depicted as a small dot at the polygon's

centroid. (These dots are intentionally small and may be very difficult to see; they are not visible in Figure 43).

- A trout: The simulated trout appear as triangle-like symbols (NetLogo's default shape) that have a color representing their species, and length proportional to the trout's simulated length. Trout are displayed at a random location within the cell they occupy.
- A redd: Redds appear as a "target" symbol of concentric circles, also colored by species.

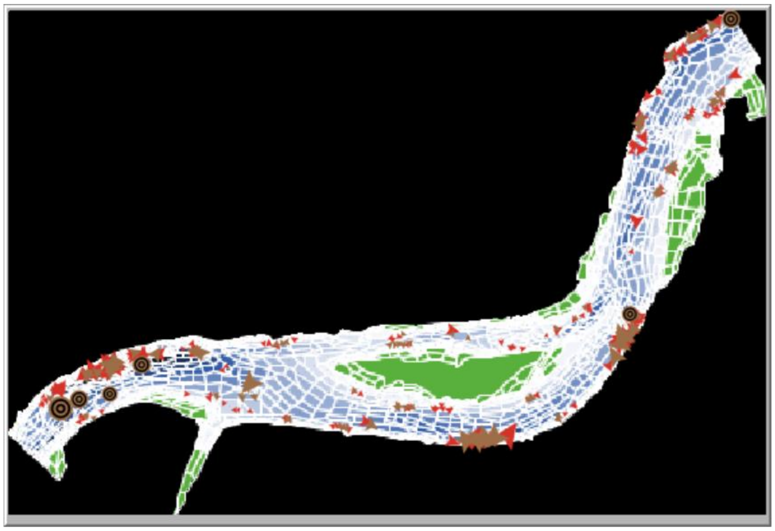


Figure 43. NetLogo's View during a simulation. White boundaries delineate cells. Here, cells are shaded by water depth. Trout appear as triangle-like shapes. There are several redds present, especially near the southwest end of the reach.

11.4.7 View settings

As the NetLogo tutorial indicates, clicking on the "Settings" button near the NetLogo menu produces a menu of View settings. For inSTREAM 7, *avoid changing any of these settings*, with one exception. Changing the "patch size" changes the size of the View on your display without affecting model results. Other settings are controlled by the inSTREAM code and should not be altered here; changing some settings (especially, "world wraps", which must be unchecked) will cause significant errors.

NetLogo also provides a speed controller near the top center of the Interface tab. For most NetLogo models, increasing the speed with this controller makes the model execute faster by reducing how frequently the View's graphical display is updated. However, inSTREAM's execution time is overwhelmingly dominated by computations, not graphics updates, and the speed controller has negligible effect.

The "view updates" checkbox should be checked (otherwise, the View will not be updated each time step), along with the option to update the view "on ticks" (once per time step) instead of "continuous". (The "continuous" option updates the view 30 times per second, which can let you see individual fish move among cells as they select habitat. This can be fun but is not very informative and slows execution considerably.)

11.5 NetLogo Export commands

In NetLogo's File menu is a submenu labeled "Export". This submenu provides tools that save parts of the model to files: you can save plots (the X-Y coordinates of all points currently on a plot), graphical images of the View or the whole interface, and even the model code (in HTML format that includes the text colors).

This menu also includes a command "Export world" that saves the complete current state of a model so it can be imported (via "Import world") and re-started later. This capability to save the model's state ("serialize" the model, in computer science terminology) could be very useful for inSTREAM users, but unfortunately it does not work for inSTREAM (as of NetLogo version 6.2). The export capability has not been implemented in NetLogo's "Time extension" that inSTREAM 7 depends on.

12 Output Files

12.1 General information on file output

InSTREAM 7 can produce many kinds of output file that are useful for typical analyses. However, our experience indicates that users must be able to control how much output they produce, and the types; otherwise, the model can produce far more output than we can understand or even store and analyze. Therefore, inSTREAM has the following file output features.

- All output files can be turned on or off easily, using switches described below. The main output files can even be turned on and off during a simulation.
- Users also control how often information is written to the population summary output files (and the individual fish and cell output files), via the file output frequency parameters explained in Sect. 14.1.1. These parameters can be set to produce output every time step or only periodically (e.g., daily, biweekly, monthly).
- InSTREAM 7 never overwrites output files and instead produces a unique name for each output file each time the model is executed. The name of each output file is a combination of the type of output, (when relevant) the BehaviorSpace run number (Sect. 15), and the date and time the model run started (from the computer clock). For example, the output file name `BriefPopOut-02-45-12.034_PM_27-May-2020.csv` is the brief population output file for a model run that started at 2:45:12 pm on 27 May 2020, with BehaviorSpace not used. (The file name would include `r1`, `r2`, `r3`, etc. for runs executed by BehaviorSpace.)
- The names of all files produced by one model run are the same except for the initial characters designating the type of output. Hence, it is easy to distinguish the output from different runs.
- All output files are written in CSV format for easy import into analysis software such as Excel or R.
- If you open an output file in Excel (or other software that locks files when editing them) as a model is running and writing to that file, it will cause an error. One way to look at an output file before a simulation finishes is simply to copy it and paste it into another directory (e.g., onto the desktop) and look at the copy. A second way is to pause the model run by clicking the "go" button, look at the file, close it, and re-start the simulation

by clicking go again. (If you do open an output file while the model is running and cause a run-time error, you can simply close the error dialog and hit the go button again to re-start the simulation.)

- The format of output date and time values is easily changed via simple code edits. Search the code for statements starting `set formatted-sim-time ...` and change the format code at the end of the statement[67]. For example, if you prefer dates in day-month-year format instead of inSTREAM's default month/day/year format, change the format code from `"M/d/yyyy HH:mm"` to `"d-M-yyyy HH:mm"`. **Do not change the date and time format elsewhere in the code, and be very careful with the format codes: never use YYYY instead of yyyy, and be aware that seemingly small differences (upper vs. lower case, one character vs. two) can cause major errors that are not easy to identify.** Check the NetLogo time extension's documentation for information on format codes.

Several categories of file output can be produced. The **main output files** are controlled via the switches on the model interface (Figure 42) and described in Sect. 12.2. These files produce the kind of summary results used in typical analyses.

Debugging output files (Sect. 12.3) provide details about specific submodels. They are designed for use only when needed to understand why the model produces a specific result (e.g., why the model trout never spawn), and sometimes produce extreme volumes of output. Debugging output files are turned off and on by model control parameters explained in Sect. 14.

Graphical output files can be produced in two ways. If the model control parameter ***write-frames?*** is set to `true`, the model produces a graphics image file (PNG format) of the model interface each time step; these image files can be used to produce a "movie" (e.g., in GIF format) of the model run. Producing movies from this output requires additional software and is not otherwise addressed in this document. A very simple code modification produces graphical output of only the View (the display of cells and trout) instead of the whole Interface. In the Code tab, search for the command `export-interface` (in the procedure `update-output`) and change it to `export-view`.

Images of the NetLogo interface can also be created manually via the "export" commands available under the File menu. These commands allow users (usually after pausing the simulation via the go button) to write individual image files of the interface or its components. The export capabilities are explained in the Interface Guide section of the NetLogo User Manual.

(The inSTREAM 7 software also includes code to write numerous output files designed specifically for testing the software, discussed in Sect. 10.3. These test output files are not addressed here because they are not useful for typical users.)

The BehaviorSpace simulation experiment tool described in Sect. 15 (see also Sect. 16.6) produces its own output instead of, or in addition to, the files described here.

[67] There is one of these `set formatted-sim-time` statements in the procedure `set-up-time` and two in `update-time-and-habitat`.

12.2 Main output files

Here we describe the main output files. These files are easily modified; standard output from future versions of inSTREAM may not exactly match this description. These files include a column containing the BehaviorSpace run number (Sect. 15) so output be associated with the parameter values used in a BehaviorSpace experiment; the run number is set to 0 when BehaviorSpace is not used.

The **brief population output** file is controlled by the interface switch labeled `brief-pop-output`. When this switch is on, the file reports the status of the simulated trout population at time intervals specified by the output frequency control parameters (Sect. 14.1.1). It also automatically produces output for all census dates, which are also specified by the output frequency control parameters. The file reports trout abundance, mean length, mean weight, mean condition, and fraction of trout using each of the three activities: drift feeding, search feeding, and hiding. These results are categorized by habitat reach, species, and age class. (Age classes are also easily defined via parameters; Sect. 14.1.5.) The file also reports the current light phase (dawn, day, dusk, night) and whether it is a census date.

The **detailed population output** file, controlled by the switch `detailed-pop-output`, differs from the brief population output file only by further breaking results out by activity: it reports the number and characteristics of trout that are drift feeding, search feeding, and hiding. It therefore is three times longer than the brief population output file.

The **redd summary output** file is controlled by the `redd-output?` switch. It writes one line of output summarizing the fate of each redd. The output line for a redd is written on the time step when the redd is emptied of eggs and removed from the model, so the file never contains information on redds that still exist. The file reports each redd's species, reach name and cell identification number, dates created and emptied, initial number of eggs, number of eggs that died of each mortality source, and number of eggs hatched into new trout.

The **fish events** and **redd events output** files are controlled by the `events-output?` switch. These two files report when specific events happen to individual trout and redds; they report all such events and are not affected by the output frequency control parameters. The fish events output contains one line for each time a trout experiences an event such as being initialized at the start of a run, emerging from a redd, dying, and spawning. This output line reports the trout's unique identification number (so the events happening to a particular individual can be tracked), and the event type (e.g., the mortality source causing the fish to die). The output also includes the date and time, the reach and cell where the event occurred, and the trout's characteristics such as age, length, weight, and condition. For redds, the events reported include being created, losing eggs to the various egg mortality sources, and when emergence of new trout starts and ends.

12.3 Debugging output files

We recommend using these output files (Table 23) only when necessary to understand or test specific parts of the model or to obtain more evidence for why specific results were produced. The first two of these files, individual fish and cell output, are updated at time intervals specified by the output frequency control parameters (Sect. 14.1.1). The others are always updated every time step.

The parameter ***frac-trout-to-output*** (a number between 0.0 and 1.0) controls the completeness of the individual fish output that allows analysis of growth and behavior of individuals over time. A ***frac-trout-to-output*** value of 1.0 produces output for every trout. For ***frac-trout-to-output*** < 1.0, each trout when created (via initialization, or during a simulation) has a probability equal to ***frac-trout-to-output*** of being tracked over its entire life. High values of ***frac-trout-to-output*** can yield very large individual fish output files, but low values (e.g., 0.1) may not provide enough data in simulations of populations with low survival to adulthood.

Table 23. Debugging output file information.

Output switch parameter	Entities reported	Output variables	Notes
`individual-fish-output?`	Each of a randomly selected fraction of the trout population; trout are selected when created, so individuals can be followed over time.	Location (reach, cell), species, age, length, weight, condition, superindividual representation, activity, and current growth rate and survival probability.	Does not produce output on the first time step. The parameter ***frac-trout-to-output*** (a number between 0.0 and 1.0) sets the percentage of trout reported. Use this file to understand what happens to individuals. Growth and survival are for the current time step (g growth; probability of surviving the time step), *not* daily values.
`cell-output?`	Cells: one line per cell.	All cell variables, plus the number of trout of each age class occupying the cell.	Cells are not reported if dry at the current flow.
`debug-growth?`	Trout and cell	All variables of a trout and the cell it is considering that affect trout growth, and the growth rate the trout would experience in the cell.	Produces extremely large files because it reports a line each time each trout calculates growth for a cell; the code for this file is normally commented out.
`debug-survival?`	Trout	Trout characteristics (species, age, length, etc.), and the daily probability of surviving each mortality source.	Results are for the cell and activity selected by the trout on the current time step.

debug-redd-survival?	Redds	Redd species and reach; habitat variables affecting redd mortality (temperature, flow, depth); and daily survival probabilities for each redd mortality source.	Results are for the cell occupied by the redd.
debug-survival-logistics?	Trout, redds	Values of the logistic functions used to model trout survival.	This output includes a table of logistic function output for inputs ranging from 0 to 50, produced only once at model initialization. It is useful only for testing and observing the logistic functions.
debug-cell-variables?	Cells	All static cell variables, including the depth and velocity lookup tables.	The output is produced only once, at model initialization.
debug-cell-hydraulics?	Cells	The depth and velocity vs. flow lookup tables.	The output is produced only once, at model initialization.
debug-cell-resources?	Cells	Cell variables representing resources depleted by trout: drift and search food, hiding places, velocity shelter.	Output is produced when variables are re-set at the start of a time step and each time a trout occupies the cell and depletes resources.
debug-spawning-readiness?	Trout	All trout and habitat variables affecting whether a trout is ready to spawn on the current time step; whether the trout spawns; and (if not) the criterion that was not met.	Output is produced once per female trout per day.

13 Input Files

InSTREAM 7 requires four kinds of site-specific input files. The first three of these types specify the model’s initial conditions (Sect. 7) and static habitat variables (Sect. 4.1), and the fourth provides the time-series inputs that drive habitat dynamics (Sect. 8.1). This section describes the file types and formats, providing first general information and then the details of each file type.

Information and recommendations on developing the input for a study site are provided in sects. 18 through 22.

The troubleshooting guide of Sect. 17 lists many of the problems that arise when assembling input, with solutions. Users must expect to encounter many error statements, but most are due to common, easily fixed problems.

13.1 Common characteristics of input files

Input files, including the parameter file, can be in the same directory as the project's NetLogo file, or in a subdirectory. When files are not in the same directory as the NetLogo file, their file names must include the path to them. This path is relative to the directory containing the NetLogo file (not the parameter file). For example, if the main project directory that contains the NetLogo file is `C:\Users\...\InSTREAM-Work\Flow-Experiment-Project` and the parameter file and input files are in the subdirectory `C:\Users\...\InSTREAM-Work\Flow-Experiment-Project\Input`, then the parameter file name must be designated in the inSTREAM NetLogo code as:

```
__includes [ "Input/parameters-Flow-Experiment.nls" ]
```

and the parameter file must include `Input/` as the first part of each input file's name. If the GIS shapefile is in its own subdirectory of the input directory, then its name must be specified as `"Input/Shapefile/Project.shp"`.

InSTREAM treats file names as case-insensitive: it does not matter if an input file is named VELOCITY_Input.csv or Velocity_Input.csv. However, text variables contained in these files *are* case-sensitive. Such variables include cell identifier codes and reach names; if the GIS file contains cells named "ReachA-Cell001", etc., and the hydraulic input file names cells as "REACHA-CELL001", inSTREAM will not recognize these as the same cells.

Except for the GIS shapefile, all the input files for inSTREAM use CSV format and are designed to be created and edited in spreadsheet software. This approach offers convenience, but users need to be aware that Excel frequently adds extra blank cells and lines when saving files in CSV format. When inSTREAM 7 raises an error when trying to read a CSV input file, the problem can often be fixed by opening the file in a text editor (e.g., Notepad++) instead of Excel and finding and removing extra commas. In a text editor, first go to the end of the file and delete any lines containing only commas. Then use the "find and replace" function to remove any commas with nothing between them (", , ,") at the ends of lines. (However, extra commas at the end of the file's header lines are harmless.)

When saving an input file from Excel, select Excel's "CSV (Comma delimited)" file type. Other CSV formats offered by Excel (e.g., "CSV-UTF-8"; see the Project's note about UTF-8 in Sect. 17) may not be readable by inSTREAM.

13.2 GIS shapefile and background image file

The shapefile (which is actually a subdirectory of about 8 files of different types) contains the spatial information about habitat cells: their location, shape, size, and characteristics. The name of the shapefile is provided via the model parameter ***GIS-file-name*** (Sect. 14.1.2).

We have found that GIS analysis is almost always necessary to prepare habitat information, and NetLogo's GIS extension makes it easy to import the information directly from a shapefile into the trout model. Additional information on how NetLogo uses GIS input is in its User Manual section on the GIS extension.

The GIS input to inSTREAM must have these characteristics:

- The format is an ESRI shapefile containing a vector dataset.
- Each "feature" of the dataset is a polygon representing one habitat cell.
- Distance units are meters.
- The projection is one of those supported by the GIS extension; these are listed in the User Manual. (We typically use shapefiles saved by the GIS in the Lambert_Conformal_Conic projection, while the GIS extension accepts only the Lambert_Conformal_Conic_2SP projection. Doing so requires editing the shapefile's .prj file to change `"Lambert_Conformal_Conic"` to `"Lambert_Conformal_Conic_2SP"`.)
- Each polygon feature must have properties for each of the cell variables listed in Table 24. (Any other properties are ignored.)

InSTREAM 7 can represent multiple reaches; different study sites can be simulated at once as if they were connected (Sect. 4.1.2). Reaches must be assembled in GIS and all included in the same shapefile (Figure 44), even though each reach has its own flow, temperature, and turbidity input. The GIS input for each reach can be assembled in separate shapefiles and then moved into a single shapefile that retains the geometry of each reach while the distance between reaches becomes artificial.

The distance between any two cells is calculated directly from the shapefile, so how multiple reaches are assembled in it can affect results. Figure 44 illustrates three reaches placed very close to each other, so even very small fish could move among reaches (Sect. 9.13.1). Placing the reaches further apart would limit movement among them to larger individuals.

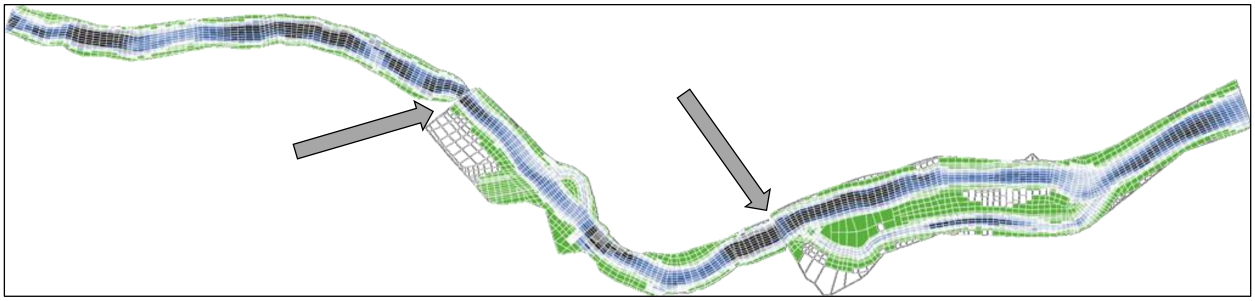


Figure 44. The View of an inSTREAM 7 application with three reaches. The arrows indicate where separate study sites were joined in GIS to produce one shapefile of cell polygons representing the reaches as if they were adjacent. The actual study sites are several km apart. (This input set has 5358 cells and represents about 3000 m of a large river.)

As explained in Sect. 14.1.2, the shapefile can use any names for the properties used by inSTREAM; the parameters listed in the second column of Table 24 are used to tell inSTREAM which shapefile properties contain which cell variables.

To avoid unnecessary computations, the inSTREAM code automatically deletes any cells that would always be dry. It checks the depth input file (Sect. 13.3) and deletes any cell that has zero depth at all flows in that file.

InSTREAM 7 has an option to provide a background image—typically, an aerial photograph of the study site—that is displayed behind the other graphics in the View (Figure 45). This image is provided as an image file (Sect. 14.1.2) in BMP, JPG, GIF, or PNG format. The name of the file is provided via the model parameter ***background-image-file***. If no image is to be used, simply set this parameter to "`None`". The image's center is placed at the center of the View and the image stretched to match the extent of the GIS shapefile. Therefore, the background image should have exactly the same spatial extent as the shapefile; normally it is prepared in GIS. (The background image is displayed using NetLogo's `import-drawing` primitive; the NetLogo documentation for that command provides details on image formats.)

Table 24. Cell variables required as polygon properties in the shapefile. The second column is the inSTREAM parameter that must contain the property name used in the shapefile. Cell variables are fully defined in Sect. 4.1.3.

Property	Parameter containing property name	Value and units
Cell identification code	`GIS-property-for-cell-ID`	A unique text code used as the cell's identifier. The same code must be used in the hydraulic input files. Multiple cells cannot have the same value of this variable even if in different reaches. Typically the code combines the reach name with a cell number, e.g., "UpperReach-Cell1093". *This property must be in text, not number, format*.
Reach name	`GIS-property-for-reach-name`	The name (text) of the cell's reach. The reach name must exactly match that used in the habitat parameters (Sect. 14).
Area	`GIS-property-for-cell-area`	The cell's area (m^2; converted by inSTREAM to cm^2).
Distance to escape cover	`GIS-property-for-cell-dist-escape`	A characteristic distance (m; converted by inSTREAM to cm) that an adult trout in the cell must move to reach escape cover (Sect. 9.18.5).
Number of hiding places	`GIS-property-for-cell-num-hiding-places`	The number of places in the cell where an adult trout could use the hiding activity (Sect. 9.18.6).
Fraction providing velocity shelter	`GIS-property-for-cell-frac-vel-shelter`	The fraction (0.0 to 1.0) of cell area providing velocity shelter for drift feeding (sects. 9.11, 9.13, 22.2).
Fraction providing spawning gravel	`GIS-property-for-cell-frac-spawn`	The fraction (0.0 to 1.0) of cell area providing gravel spawning habitat (Sect. 9.29).

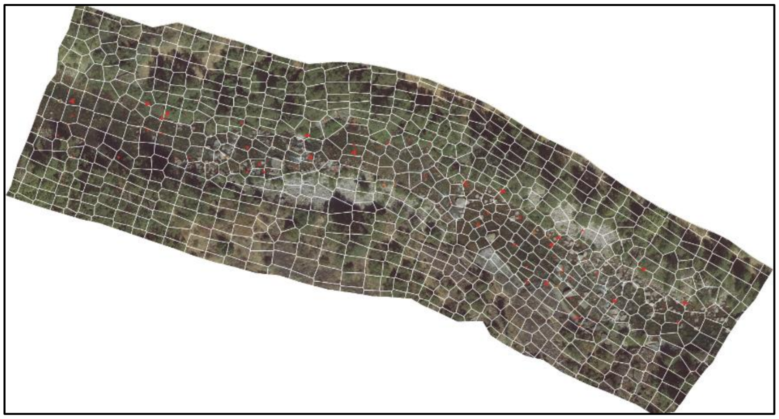


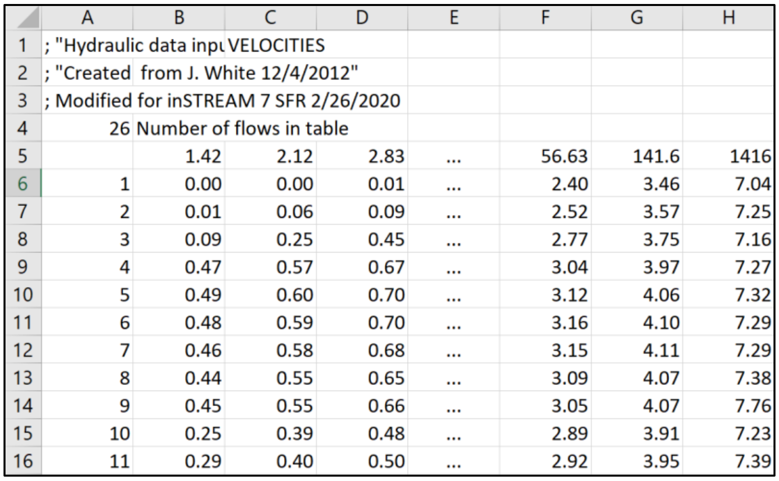
Figure 45. Example View with the optional background image. The image was trimmed in GIS to the same extent and shape as the habitat shapefile.

13.3 Hydraulic input

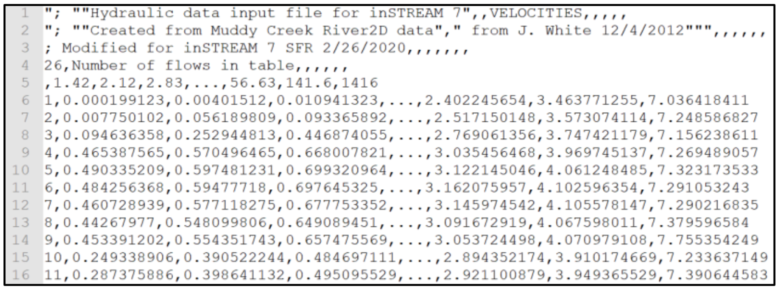
The hydraulic input consists of two files for each reach; these files contain lookup tables that relate the water depth and velocity in each cell of a reach to the reach's flow rate. The names of these files are provided as the reach parameters ***depth-file-name*** and ***velocity-file-name*** (Sect. 14.2). Preparation of these lookup tables is addressed in Sect. 21. The depth and velocity tables are provided in separate input files that have identical format. The files are in CSV format and designed to be edited in spreadsheet software.

The hydraulic input files (illustrated in Figure 46) contain:

- An arbitrary number of lines of header information and comments that are ignored by inSTREAM. These header lines must start with a semicolon. We recommend that they document the reach the file represents, whether the file contains depths or velocities, and who prepared the file and when.
- A row that contains, in its first column, the number of flows in the lookup table.
- A row that starts with an empty cell and then contains each of the flows. These flows must be in ascending order and in units of m^3/s. (In the file shown in Figure 46, the velocity lookup table contains 26 flows, ranging from 1.42 to 1416 m^3/s.)
- One row for each cell in the reach. The row starts with the cell identification code, which must match the code used in the GIS shapefile (Sect. 13.2). The next columns contain the depth or velocity that the cell experiences at the corresponding flow (the flow in the same column of row 5). For the depth input file, depths must be in m; velocity input must be in m/s. These values are converted to inSTREAM's standard length unit of cm when read in. For example, in Figure 46 cell 10 has a velocity of 0.25 m/s when the reach's flow is 1.42 m^3/s, a velocity of 0.39 m/s at a flow of 2.12 m^3/s, etc., up to 7.23 m/s at 1416 m^3/s.



	A	B	C	D	E	F	G	H
1	; "Hydraulic data inpu		VELOCITIES					
2	; "Created	from J. White 12/4/2012"						
3	; Modified for inSTREAM 7 SFR 2/26/2020							
4	26	Number of flows in table						
5		1.42	2.12	2.83	...	56.63	141.6	1416
6	1	0.00	0.00	0.01	...	2.40	3.46	7.04
7	2	0.01	0.06	0.09	...	2.52	3.57	7.25
8	3	0.09	0.25	0.45	...	2.77	3.75	7.16
9	4	0.47	0.57	0.67	...	3.04	3.97	7.27
10	5	0.49	0.60	0.70	...	3.12	4.06	7.32
11	6	0.48	0.59	0.70	...	3.16	4.10	7.29
12	7	0.46	0.58	0.68	...	3.15	4.11	7.29
13	8	0.44	0.55	0.65	...	3.09	4.07	7.38
14	9	0.45	0.55	0.66	...	3.05	4.07	7.76
15	10	0.25	0.39	0.48	...	2.89	3.91	7.23
16	11	0.29	0.40	0.50	...	2.92	3.95	7.39



```
"; ""Hydraulic data input file for inSTREAM 7",,VELOCITIES,,,,,
"; ""Created from Muddy Creek River2D data"," from J. White 12/4/2012""",,,,,,
; Modified for inSTREAM 7 SFR 2/26/2020,,,,,,,
26,Number of flows in table,,,,,,
,1.42,2.12,2.83,...,56.63,141.6,1416
1,0.000199123,0.00401512,0.010941323,...,2.402245654,3.463771255,7.036418411
2,0.007750102,0.056189809,0.093365892,...,2.517150148,3.573074114,7.248586827
3,0.094636358,0.252944813,0.446874055,...,2.769061356,3.747421179,7.156238611
4,0.465387565,0.570496465,0.668007821,...,3.035456468,3.969745137,7.269489057
5,0.490335209,0.597481231,0.699320964,...,3.122145046,4.061248485,7.323173533
6,0.484256368,0.59477718,0.697645325,...,3.162075957,4.102596354,7.291053243
7,0.460728939,0.577118275,0.677753352,...,3.145974542,4.105578147,7.290216835
8,0.44267977,0.548099806,0.649089451,...,3.091672919,4.067598011,7.379596584
9,0.453391202,0.554351743,0.657475569,...,3.053724498,4.070979108,7.755354249
10,0.249338906,0.390522244,0.484697111,...,2.894352174,3.910174669,7.233637149
11,0.287375886,0.398641132,0.495095529,...,2.921100879,3.949365529,7.390644583
```

Figure 46. Example extract from a hydraulic input file edited in (top) Excel and (bottom) a text editor. (The ellipses in column E do not appear in the file but instead represent the 20 columns not shown in this figure.)

13.4 Initial population characteristics

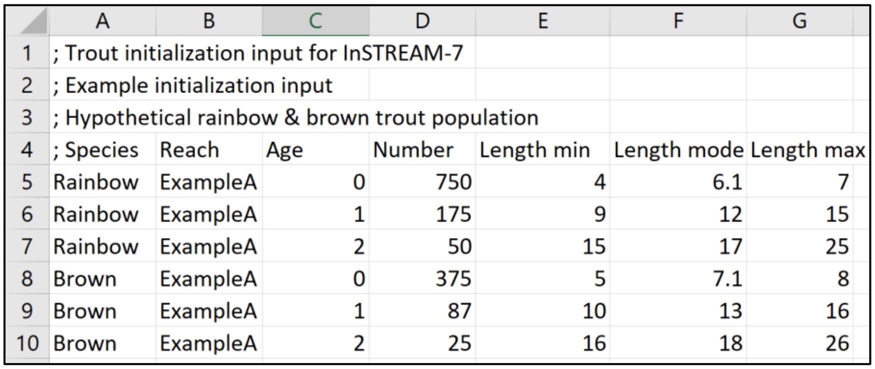
The initial population file provides the information needed to create the initial trout population, using methods described in Sect. 7.3. The name of the file is provided as the model parameter ***initial-population-file*** (Sect. 14.1.2).

This input file starts with an arbitrary number of header and comment lines that should include the column headers (row 4 in Figure 47). These header and comment lines must start with a semicolon: the model software ignores any line in this file that either starts with a semicolon or is completely blank. (The NetLogo programming language uses semicolons to designate comments.)

The remaining lines each hold the information to initialize one age class of one species in one reach. These lines must contain the following values, in the exact order shown in Figure 47.

- The species name, exactly as in the parameters that define which species are being simulated (Sect. 14.3). This value is case-sensitive.
- The name of the reach the trout are to be initialized in. The reach name (also case-sensitive) must exactly match the reach's name in the parameters defining reaches (Sect. 14.1.7) and in the GIS file.

- The trout age, an integer. These ages do not have to correspond to the age classes used to summarize output (Sect. 14.1.5): for example, even if output uses age classes of 0, 1, and 2 and older, this input can specify ages of 1, 2, 3, and 4.
- The number of trout to be initialized.
- The minimum, mode, and maximum of the triangular distribution used to assign lengths to the initial trout.



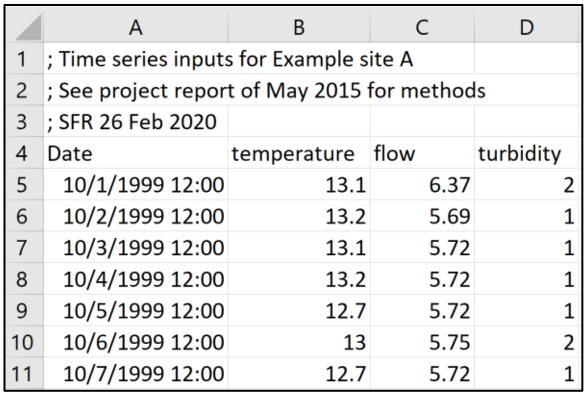
	A	B	C	D	E	F	G
1	; Trout initialization input for InSTREAM-7						
2	; Example initialization input						
3	; Hypothetical rainbow & brown trout population						
4	; Species	Reach	Age	Number	Length min	Length mode	Length max
5	Rainbow	ExampleA	0	750	4	6.1	7
6	Rainbow	ExampleA	1	175	9	12	15
7	Rainbow	ExampleA	2	50	15	17	25
8	Brown	ExampleA	0	375	5	7.1	8
9	Brown	ExampleA	1	87	10	13	16
10	Brown	ExampleA	2	25	16	18	26

Figure 47. Example initial population initialization input, viewed in Excel.

13.5 Time series inputs

The time series input file provides the values of the habitat variables—flow, temperature, and turbidity—that vary over time (Figure 48). See the Input Data section of the model description (Sect. 8.1) to understand exactly what input is needed and how it is used (e.g., whether input must be daily, what time to assign to daily mean values). More than one reach can share the same time series file, or each reach can have its own. The names of these files are provided as the reach parameter ***time-series-input-file*** (Sect. 14.2). This CSV file must contain:

- A variable number of header and comment lines, which must each start with a semicolon.
- A row of column labels, not preceded by a semicolon. The first label ("Date" in Figure 48) is ignored by the software because the first column of input (cells A5 to A11 in Figure 48) must be the date and time. The remaining column labels must include the exact words "temperature", "flow", and "turbidity", all lower-case and with no spaces. (Accidentally including a space in these column labels will cause a run-time error.) The order in which temperature, flow, and turbidity appear in the file does not matter; flow could be in column B, for example, but date and time must always be in the first column.
- The input rows providing flow, temperature, and turbidity for each date and time. The first column must be the date and time in M/d/yyyy h:mm format. (In Excel, this format is found under "Custom" in the number format menu. Be aware that Excel often changes the data-time format when opening a CSV file, so you must check that the date and time values in column A use the correct format before saving the file for use in inSTREAM.)



	A	B	C	D
1	; Time series inputs for Example site A			
2	; See project report of May 2015 for methods			
3	; SFR 26 Feb 2020			
4	Date	temperature	flow	turbidity
5	10/1/1999 12:00	13.1	6.37	2
6	10/2/1999 12:00	13.2	5.69	1
7	10/3/1999 12:00	13.1	5.72	1
8	10/4/1999 12:00	13.2	5.72	1
9	10/5/1999 12:00	12.7	5.72	1
10	10/6/1999 12:00	13	5.75	2
11	10/7/1999 12:00	12.7	5.72	1

Figure 48. Example time series input file, viewed in Excel.

While this file was designed to be maintained in spreadsheet software, be aware that software such as Excel often puts quotation marks around text values when saving in CSV format. Currently, for the time series input file, quotation marks around header line text cause inSTREAM to not recognize the semicolons that tell the software to ignore the line. If inSTREAM raises an error when reading this file, open it in a text editor and remove any quotation marks. This file is also particularly likely to have blank lines left in it by Excel, which also cause errors and must be removed in a text editor (Sect. 13.1).

If you prefer to use a different date and time format in the first data column, it can be changed with a simple edit of the inSTREAM code. In the `set-up-time` procedure, find the statement that includes `ts-load-with-format` and edit its format code; for example you can change the default format from "`M/d/yyyy H:mm`" to "`d-M-yyyy H:mm`" to read input such as `18-8-2012 12:00`. **Do not change the date and time format elsewhere in the code, and be very careful with the format codes: never use YYYY instead of yyyy, and be aware that seemingly small differences (upper vs. lower case, one character vs. two) can cause major errors that could go unnoticed.** Check the NetLogo time extension's documentation for information on format codes.

14 Simulation Control and Parameter Values

Like all models, inSTREAM 7 uses parameters that are controllable by the user but constant during a model run. In addition to the model, habitat, and trout parameters (these terms are defined in Sect. 2.2.2), we refer here also to simulation control parameters that determine what inputs are used and what outputs produced.

InSTREAM 7 uses a parameter file: a separate file of NetLogo code that contains only the statements that set parameter values. The parameter file is named something like `parameters-MyProject.nls` and normally resides in the main project directory. This file is executed by the inSTREAM code upon model initialization. You will need to understand the following about the parameter file:

- This file is always customized to a project, so it should be given a project-specific name. The file can have any name but must use the extension `.nls`.
- It is good practice to add comments at the top of the parameter file and elsewhere in it to document the scenario(s) it represents and the changes made to it. Any line that starts

with a semicolon (or any text following a semicolon anywhere) is treated as a comment and ignored by the computer.

- The inSTREAM NetLogo file must be given the name of the parameter file. This is done near the very top of the NetLogo file's Code tab in a statement like:

  ```
  __includes [ "parameters-MyProject.nls" ]
  ```

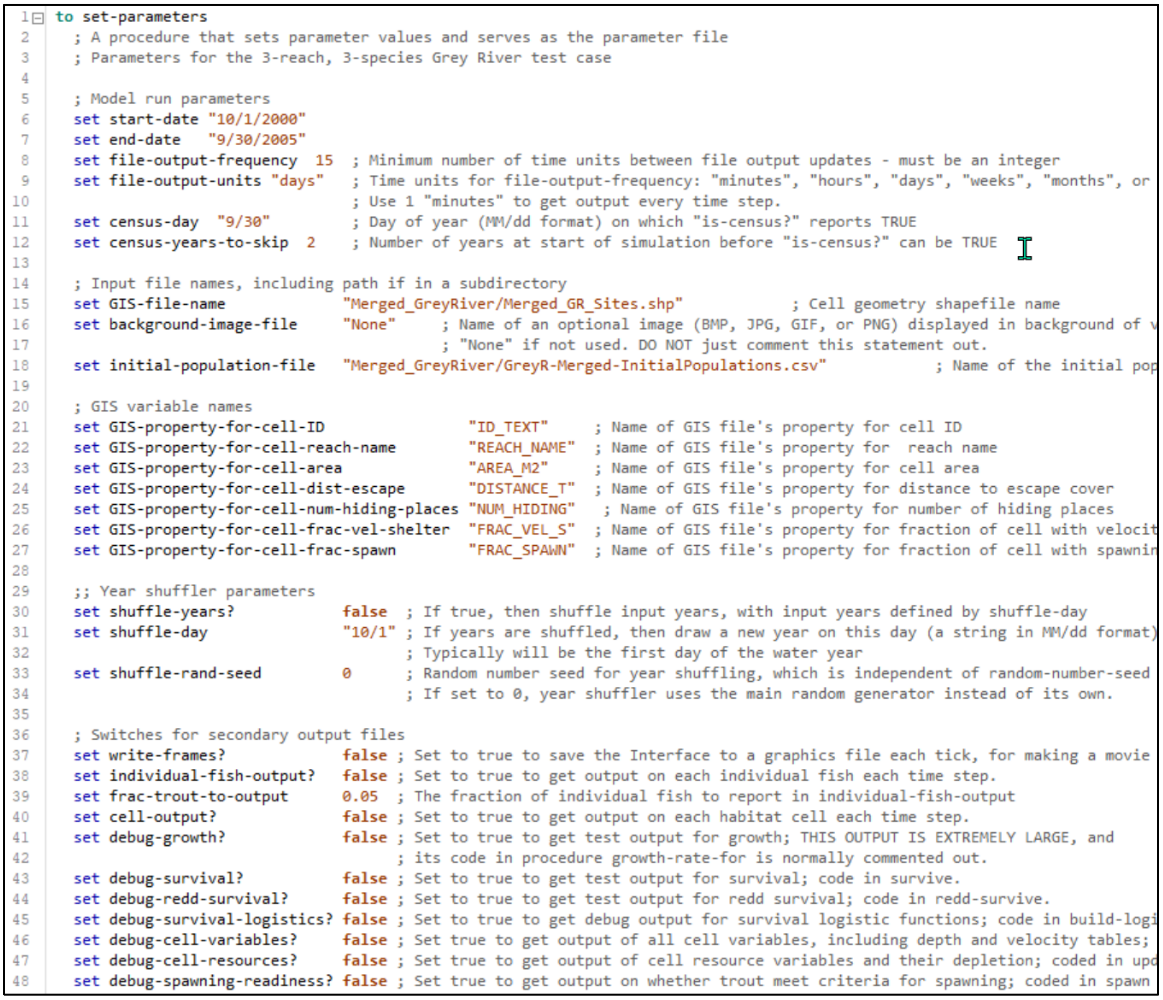
  This statement means that the statements in the parameter file are included in the model's code. Simply edit this line as needed to provide the correct parameter file name and then save the NetLogo file.
- If the file name is incorrect in the `__includes` statement, NetLogo will display a yellow error message at the top of the Code tab saying that it could not find the file. If that happens, correct the file name and click the "Check" button to make NetLogo re-check the code and clear the error statement.
- The parameter file can be opened and edited from the Code tab by clicking the tab's rectangular button "Included files". That button opens a menu that lets you select the parameter file. After making changes, you must use NetLogo's File menu to save the parameter file.
- You can also edit the parameter file in a text editor; it is a plain text file. However, this must be done before opening the inSTREAM NetLogo file.
- NetLogo's handling of included files requires that users, after editing the parameter file, **save and close it, and close the entire NetLogo file**, and then re-open the inSTREAM NetLogo file. Otherwise, changes made to parameter values may not be used in the next model run. This procedure is necessary for NetLogo versions 6.x, but the non-intuitive NetLogo behavior that necessitates it may be corrected in later versions.
- The parameter file consists mostly of `set` statements that each assign a value to a parameter, e.g.: `set start-date "10/1/2000"` (Figure 49). The order in which these set statements appear in the file has no effect. (The table input described in Sect. 14.3 is an exception: its statements must be in a specific order.)
- Text input such as file names and dates must be in quotation marks. However, many model control parameters are Boolean variables with values of either `true` or `false`, without quotation marks.

The following subsections show what the parameter file looks like and list the parameters, by category, with information on their format and values. All parameters (except some of the input parameters) are fully defined in the model description of Part II.

In the following parameter tables, grey shading indicates site-specific parameters that deserve re-evaluation for new sites.

14.1 Model parameters

These parameters (called "Observer" parameters in NetLogo terminology) control model execution, provide input file names, control the kinds of output produced, and define a few global processes. They normally appear at the top of the parameter file (Figure 49). Each group of model parameters is discussed separately in the following subsections.



```
to set-parameters
  ; A procedure that sets parameter values and serves as the parameter file
  ; Parameters for the 3-reach, 3-species Grey River test case

  ; Model run parameters
  set start-date "10/1/2000"
  set end-date   "9/30/2005"
  set file-output-frequency  15  ; Minimum number of time units between file output updates - must be an integer
  set file-output-units "days"   ; Time units for file-output-frequency: "minutes", "hours", "days", "weeks", "months", or
                                 ; Use 1 "minutes" to get output every time step.
  set census-day  "9/30"         ; Day of year (MM/dd format) on which "is-census?" reports TRUE
  set census-years-to-skip  2    ; Number of years at start of simulation before "is-census?" can be TRUE

  ; Input file names, including path if in a subdirectory
  set GIS-file-name             "Merged_GreyRiver/Merged_GR_Sites.shp"            ; Cell geometry shapefile name
  set background-image-file     "None"     ; Name of an optional image (BMP, JPG, GIF, or PNG) displayed in background of v
                                           ; "None" if not used. DO NOT just comment this statement out.
  set initial-population-file   "Merged_GreyRiver/GreyR-Merged-InitialPopulations.csv"            ; Name of the initial pop

  ; GIS variable names
  set GIS-property-for-cell-ID                 "ID_TEXT"     ; Name of GIS file's property for cell ID
  set GIS-property-for-cell-reach-name         "REACH_NAME"  ; Name of GIS file's property for  reach name
  set GIS-property-for-cell-area               "AREA_M2"     ; Name of GIS file's property for cell area
  set GIS-property-for-cell-dist-escape        "DISTANCE_T"  ; Name of GIS file's property for distance to escape cover
  set GIS-property-for-cell-num-hiding-places "NUM_HIDING"   ; Name of GIS file's property for number of hiding places
  set GIS-property-for-cell-frac-vel-shelter  "FRAC_VEL_S"   ; Name of GIS file's property for fraction of cell with velocit
  set GIS-property-for-cell-frac-spawn         "FRAC_SPAWN"  ; Name of GIS file's property for fraction of cell with spawnin

  ;; Year shuffler parameters
  set shuffle-years?            false  ; If true, then shuffle input years, with input years defined by shuffle-day
  set shuffle-day               "10/1" ; If years are shuffled, then draw a new year on this day (a string in MM/dd format)
                                       ; Typically will be the first day of the water year
  set shuffle-rand-seed         0      ; Random number seed for year shuffling, which is independent of random-number-seed
                                       ; If set to 0, year shuffler uses the main random generator instead of its own.

  ; Switches for secondary output files
  set write-frames?             false ; Set to true to save the Interface to a graphics file each tick, for making a movie
  set individual-fish-output?   false ; Set to true to get output on each individual fish each time step.
  set frac-trout-to-output      0.05  ; The fraction of individual fish to report in individual-fish-output
  set cell-output?              false ; Set to true to get output on each habitat cell each time step.
  set debug-growth?             false ; Set to true to get test output for growth; THIS OUTPUT IS EXTREMELY LARGE, and
                                      ; its code in procedure growth-rate-for is normally commented out.
  set debug-survival?           false ; Set to true to get test output for survival; code in survive.
  set debug-redd-survival?      false ; Set to true to get test output for redd survival; code in redd-survive.
  set debug-survival-logistics? false ; Set to true to get debug output for survival logistic functions; code in build-logi
  set debug-cell-variables?     false ; Set true to get output of all cell variables, including depth and velocity tables;
  set debug-cell-resources?     false ; Set true to get output of cell resource variables and their depletion; coded in upd
  set debug-spawning-readiness? false ; Set true to get output on whether trout meet criteria for spawning; coded in spawn
```

Figure 49. Example parameter file, part 1: Model run control, input file, GIS variable name, year shuffler, and secondary output control parameters. This and subsequent figures illustrate what the file looks like in NetLogo's code editor.

14.1.1 Model run control

These parameters (Table 25) control the time period simulated and when some outputs are produced. The output frequency and "census" parameters determine when the following output files (Sect. 12) are updated: brief population output, detailed population output, individual fish, and cell output. When activated, all other outputs are updated each time step.

The output frequency parameters specify the minimum time between updates of these output files. Output is always updated on the first time step of a simulation. The parameter ***file-output-frequency*** specifies how many time units are simulated between output updates, and ***file-output-units*** specifies what time units are used; valid values include `"minutes"`, `"hours"`, `"days"`, and `"months"`. For example, if ***file-output-frequency*** is set to 1 and ***file-output-units*** is set to `"months"`, then the population output files will report the population status once per month, starting on the simulation's first time step. To produce output every time step, set ***file-output-frequency*** to `1` and ***file-output-units*** to `"minutes"`. (The model never updates its output more than once per time step.)

The parameters ***census-days*** and ***census-years-to-skip*** are designed to make it easy to compare population file output to field data that were collected at the same times each year. Trout populations are often censused in early and late summer; setting ***census-days*** to `(list "5/1" "9/30")` will force inSTREAM to produce output on the first time step of May 1 and September 30 each year and label that output as a census day, whether or not output would normally be produced then. Setting ***census-years-to-skip*** to 3 years, for example, tells inSTREAM not to produce this census output until three full years have been simulated, to reduce the effects of initial population status on results. Therefore, filtering the output to include only census results (`is-census?` has a value of `true` in the population output files) identifies results for comparison to the late-summer field data. The ***census-days*** list can have as many values as desired, including none (in which case `is-census?` is never true); ***census-years-to-skip*** can be zero.

(A simple code change can cause output to be produced *only* on census days, if desired[68].)

Table 25. Model run control parameters.

Parameter	Value type	Example value	Parameter function
`start-date`	Date in M/d/yyyy format	`"10/1/2000"`	The first date simulated.
`end-date`	Date in M/d/yyyy format	`"9/30/2005"`	The last date simulated.
`file-output-frequency`	Integer	`2`	With `file-output-units`, the minimum time between output of the population output files. Use `1 "minutes"` to get output each time step.
`file-output-units`	Text	`"weeks"`	Time unit for `file-output-frequency`. Valid values are: `"minutes"`, `"hours"`, `"days"`, `"weeks"`, `"months"`, and `"years"`.
`census-days`	List of days of the year in M/d format	`(list "5/1" "9/30")`	Dates on which population output is forced and labeled as a census date each year.
`census-years-to-skip`	Integer	`3`	Number of years skipped at the beginning of a simulation before census output is produced.

[68] In the procedure `update-output`, find the statement that determines whether to write to a specific output file, e.g., `if brief-pop-output?` and change it to `if brief-pop-output? and is-census?`.

14.1.2 Input file names and GIS variable names

These parameters (Table 26) tell NetLogo the names and locations of input files, and which GIS shapefile property names to use for cell variables. However, not all the input files are considered here: some are unique to each habitat reach and so are provided as reach parameters (Sect. 14.1.7).

For input files stored in different directories from the main project directory, the file names must include the path to the file. The `GIS-file-name` below provides an example: the shapefile is in a subdirectory of the main project file called `ExampleSiteGIS`.

Table 26. Input file name and GIS variable name parameters.

Parameter	Value type	Example value	Parameter function
`GIS-file-name`	Text	`"ExampleSiteGIS/ExampleSite.shp"`	The name of the GIS shapefile providing cell input.
`background-image-file`	Text	`"ExampleSiteAirPhoto.jpg"`	The name of an optional image, such as an air photo of the site, that appears in the View background (Sect. 13.2). If no image is used, then use the value `"None"`.
`initial-population-file`	Text	`"ExampleSite_InitialPopulation.csv"`	The name of the initial population input file.
`GIS-property-for-cell-ID`	Text	`"ID_TEXT"`	The name of the shapefile property containing the cell identification code.
`GIS-property-for-cell-reach-name`	Text	`"REACH_NAME"`	The shapefile property containing the reach name.
`GIS-property-for-cell-area`	Text	`"AREA_M2"`	The property containing cell area.
`GIS-property-for-cell-dist-escape`	Text	`"DISTANCE_T"`	The property containing distance to escape cover.
`GIS-property-for-cell-num-hiding-places`	Text	`"NUM_HIDING"`	The property containing number of hiding places.

Parameter	Value type	Example value	Parameter function
GIS-property-for-cell-frac-vel-shelter	Text	"FRAC_VEL_S"	The property containing cell fraction with velocity shelter.
GIS-property-for-cell-frac-spawn	Text	"FRAC_SPAWN"	The property containing cell fraction with spawning gravel.

14.1.3 Year shuffler controls

The year shuffler is part of model run control, but its parameters are listed separately (Table 27) because it is less-often used. The year shuffler is a way to replicate simulations by randomly shuffling the sequence of water years in the flow, temperature, and turbidity input (Sect. 8.2).

It is important to understand how the parameter ***shuffle-rand-seed*** works. If it is set to 0, then the year shuffler uses the same random number generator as the rest of the model. Therefore, if both ***shuffle-rand-seed*** and ***random-number-seed*** are set to 0, a new random sequence of input years will be used each model run. (The parameter ***random-number-seed*** is the model's main random number seed, explained in Sect. 6.9. Its value is set on the Interface tab.) If instead ***random-number-seed*** is set to a non-zero number while ***shuffle-rand-seed*** is 0, then the same sequence of shuffled input years will be used, and running the model twice (with no other changes) would produce exactly the same results. However, if ***shuffle-rand-seed*** is given a non-zero value, the model will use same sequence of input years no matter the value of ***random-number-seed***. Changing the value of ***shuffle-rand-seed*** produces a different sequence of input years.

Therefore, there are two ways to use the year shuffler for its primary purpose, replicating simulations by shuffling the input water years. The simplest is to set both ***shuffle-rand-seed*** and ***random-number-seed*** to 0 so different random numbers are used for everything in each model run. The second way is to provide a different value of ***shuffle-rand-seed*** for each model run; we explain a way to do so in Sect. 15.

Table 27. Year shuffler parameters.

Parameter	Value type	Example value	Parameter function
shuffle-years?	Boolean (true or false)	false	If true, activates the year shuffler.
shuffle-day	Day of the year in M/d format	"10/1"	If the year shuffler is used, then a new input year starts on this day each year.

Parameter	Value type	Example value	Parameter function
`shuffle-rand-seed`	Non-negative integer	`79837`	Random number seed for year-shuffling; if zero, then the model's main random generator is used by the year shuffler.

14.1.4 Secondary output switches

These parameters turn secondary output files on or off. These files are listed and described in Sect. 12.3 and Table 23, so only examples are shown here in Table 28.

Table 28. Secondary output control parameters.

Parameter	Value type	Example value	Parameter function
`write-frames?`	Boolean (`true` or `false`)	`false`	If true, a graphics file of the Interface is written each time step.
`individual-fish-output?`	Boolean	`true`	If true, produces output reporting the state of selected individual trout.
`frac-trout-to-output`	Real number between 0.0 and 1.0	`0.05`	The fraction of trout selected for reporting in the individual fish output.
`cell-output?` etc.	Boolean	`false`	If true, activates an optional detailed output file.

14.1.5 Display and output control parameters

These parameters (Figure 50; Table 29) control the graphical display and how trout populations are summarized in file output.

One of these parameters, ***world-resolution***, deserves special attention because it has strong effects on factors other than display. NetLogo's built-in distance commands use distances in units of patch widths (patches being the built-in square grid cells that inSTREAM uses mainly for displaying cell variables; Sect. 11.4). The value of ***world-resolution*** is the patch width in cm, so it determines how many patches are needed to display the space represented in inSTREAM's GIS shapefile. This resolution must be re-evaluated for each new study site, considering its two effects:

- The lower the value of ***world-resolution***, the more patches NetLogo uses. More patches require more computations, which means slower execution (especially to initialize the model) and more memory use per model run. Therefore, ***world-resolution*** should be no lower than necessary.

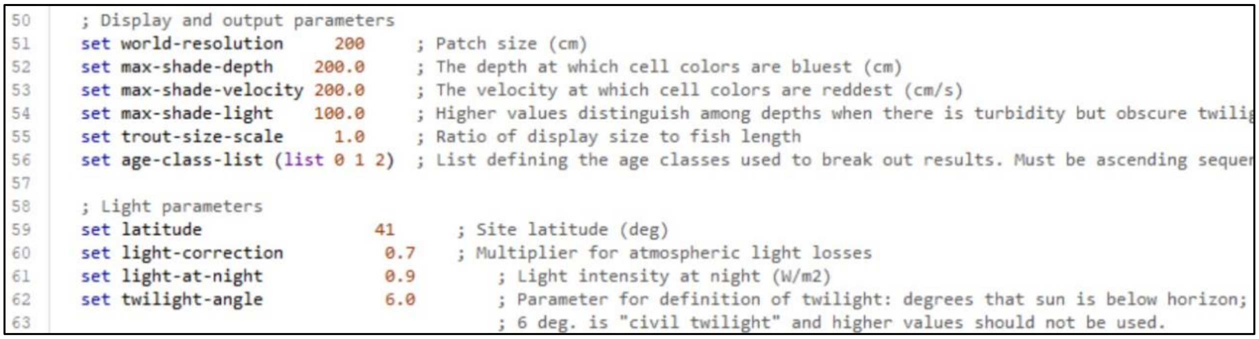
- If ***world-resolution*** is too high, inSTREAM cannot be initialized properly because each cell must have at least one unique patch and two cells cannot occupy the same patch. If this occurs, inSTREAM will raise an error when setup is executed. The error statement recommends decreasing the value of ***world-resolution*** to solve the problem. (This error is partially stochastic, so some values of ***world-resolution*** will work sometimes but not always. Also, too-small values of ***world-resolution*** can cause different run-time errors that do not clearly identify the problem; see the trouble-shooting guide of Sect. 17.)

Therefore, setting up input for a new site requires adjusting ***world-resolution*** by trial and error to find a value that is just low enough for each cell to have its own patch. However, the inSTREAM software includes a utility procedure to identify unusually small cells so their size can be used to find a value of ***world-resolution*** at which all cells contain at least one patch. (Alternatively, each such small cell can be merged with an adjacent cell to allow higher values of ***world-resolution***. This merging must be done in GIS and its changes must also be reflected in the hydraulic input files.) To use this procedure, enter `find-small-cells` in the Command Center on the NetLogo interface (you do not need to execute setup first). The procedure then reports the patch size (the square of ***world-resolution***, cm^2) and a list (including the area) of any cells with area less than one patch.

Changing the value of ***world-resolution*** also changes the size of the graphical display on the Interface tab—the View. However, those changes can be counteracted by changing the patch size in Model Settings menu (Sect. 11.4.7). For example, changing ***world-resolution*** from 100 to 50 cm doubles the size of the View but changing patch size from 3 to 1.5 pixels per patch returns the View to its original size and has no other effect. (Patch size, unlike ***world-resolution***, has no effect on computations.)

The parameters ***max-shade-depth***, ***max-shade-velocity***, and ***max-shade-light*** control how inSTREAM colors cells when the user decides to shade them by depth, velocity, or light (Sect. 11.4.3). They define the value of depth (or velocity or light) at which the shaded color reaches its most intense. For example, when ***max-shade-velocity*** is 200 cm/s the cell colors shade from yellow to red as velocity increases from zero to 200 cm/s, but cells with velocity > 200 all have the same dark red color. These variables can be changed to reflect the size of stream being modeled: if depths > 200 cm are widespread, for example, then the value of ***max-shade-depth*** could be increased.

The parameter ***age-class-list*** defines the age classes used in the population output files. (These age classes have no effect on model results, only how they are summarized in output files.) The parameter must be set to a list of integer ages, starting with 0. If its value is `(list 0 1 2)`, then the population output files report results broken out by ages 0, 1, and 2 or more years. Using more age classes requires only adding ages to this list. (This parameter does not, however, control the age classes in the population graph on the Interface tab, or in output of the BehaviorSpace experiment manager described in Sect. 15.)



```
; Display and output parameters
set world-resolution     200     ; Patch size (cm)
set max-shade-depth    200.0     ; The depth at which cell colors are bluest (cm)
set max-shade-velocity 200.0     ; The velocity at which cell colors are reddest (cm/s)
set max-shade-light    100.0     ; Higher values distinguish among depths when there is turbidity but obscure twilig
set trout-size-scale     1.0     ; Ratio of display size to fish length
set age-class-list (list 0 1 2)  ; List defining the age classes used to break out results. Must be ascending sequer

; Light parameters
set latitude                41      ; Site latitude (deg)
set light-correction         0.7    ; Multiplier for atmospheric light losses
set light-at-night           0.9        ; Light intensity at night (W/m2)
set twilight-angle           6.0        ; Parameter for definition of twilight: degrees that sun is below horizon;
                                        ; 6 deg. is "civil twilight" and higher values should not be used.
```

Figure 50. Example parameter file, part 2: Display and output, and light parameters.

Table 29. Display and output parameters.

Parameter	Value type	Example value	Parameter meaning
`world-resolution`	Positive integer	`200`	The model display's spatial resolution as patch width (cm); see the above discussion.
`max-shade-depth`	Real number	`200`	The cell depth (cm) above which cell colors are bluest when shaded by depth. (Below this depth, (Shade changes with depth.)
`max-shade-velocity`	Real number	`300`	The cell velocity (cm/s) above which cell colors are reddest (shade changes with velocity below this value), when cells are shaded by velocity.
`max-shade-light`	Real number	`100`	The cell irradiance (W/m^2) above which cell colors are whitest when shaded by light.
`trout-size-scale`	Real number	`1.0`	The ratio of trout display size to simulated length.
`age-class-list`	NetLogo list	`(list 0 1 2)`	The age classes used to summarize results in output files. The numbers must be ascending sequential integers that start with zero.

14.1.6 Light parameters

Parameters of the light-related submodels that do not to differ among reaches are treated as model parameters (Figure 50, Table 30).

Table 30. Light parameters.

Parameter	Value type	Example value	Parameter meaning
`latitude`	Real number	`38`	The site latitude (degrees north latitude).
`light-correction`	Real number between 0.0 and 1.0	`0.8`	The fraction of light lost to the atmosphere (but not to shading).
`light-at-night`	Real number	`0.9`	The water surface irradiance (W/m^2) at night.
`twilight-angle`	Real number	`6.0`	The angle (degrees) below the horizon defining the start of dawn and end of dusk; 6° is the conventional definition of twilight.

14.1.7 Input testing parameters

InSTREAM 7 has one parameter to control how it checks input files: ***max-days-between-time-series-inputs***. During model initialization, the software checks the time series input file(s) (Sect. 13.5) for missing values. Because inSTREAM 7 does not restrict input to a specific temporal resolution (Sect. 8.1), users control this check via the value of ***max-days-between-time-series-inputs***. The model software searches the input file for any places where the time between input records (rows) exceeds the value of ***max-days-between-time-series-inputs***, which is in days. (This check is made only for dates used in the simulation, between ***start-date*** and ***end-date***.)

If a model application uses daily input, then ***max-days-between-time-series-inputs*** should have a value of 1.0 d; then, the inSTREAM 7 software will notify the user of any places in the input with more than one day between records. If an application uses hourly input, then ***max-days-between-time-series-inputs*** should be set to 0.042 d (just over one hour). Input checking can be essentially turned off by setting ***max-days-between-time-series-inputs*** to a large number such as 30 d; however, missing time series input is a very common problem so we strongly recommend leaving this check active.

14.2 Reach parameters

Reach parameters (Figure 51) include both initialization information—the names of input files that must be provided for each reach—and habitat parameters that can differ among reaches.

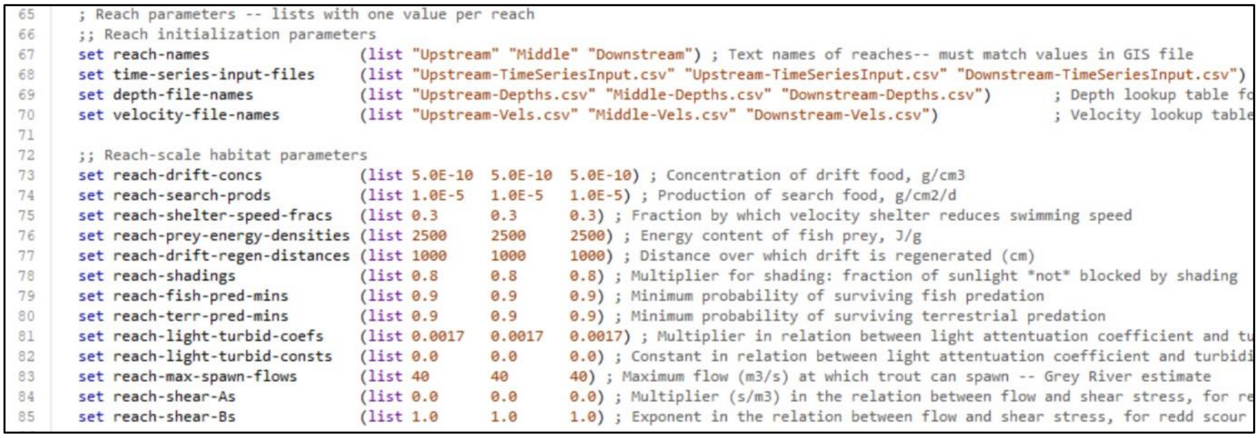
To accommodate multiple reaches, the values of these parameters are entered as NetLogo “lists”: variables that contain multiple values. The reach parameter code in the parameter file consists of statements such as:

```
set reach-names           (list "Upstream" "Middle" "Downstream")
…
set reach-drift-concs     (list 5.0E-10  3.7E-10  4.9E-10)
```

These statements mean that the project includes three reaches, and that the values of reach parameters (such as `reach-drift-concs` shown here) are entered as a list of three values assigned to the reaches named Upstream, Middle, and Downstream. The order of the values in each parameter's list determines which reach it is assigned to: in the above example, the first value in each reach parameter list will be used for the reach named "Upstream", the second value will be used for the "Middle" reach, etc. (Even if there is only one reach, the word `list` must appear before the parameter value, all inside parentheses.)

The reach parameter names in the parameter file are the plurals of the names used in the model description of Part II. Plurals are used in the parameter file because inSTREAM uses the lists of values provided there to set the parameter value of each individual reach: the list created by the statement `set reach-drift-concs` is used in the code to set the value of each reach's parameter ***reach-drift-conc***.

Note that some parameter values are likely to be in the scientific notation format used by computers. For example, `3.7E-10` means 3.7 × 10^{-10}.



```
; Reach parameters -- lists with one value per reach
;; Reach initialization parameters
set reach-names                 (list "Upstream" "Middle" "Downstream") ; Text names of reaches-- must match values in GIS file
set time-series-input-files     (list "Upstream-TimeSeriesInput.csv" "Upstream-TimeSeriesInput.csv" "Downstream-TimeSeriesInput.csv")
set depth-file-names            (list "Upstream-Depths.csv" "Middle-Depths.csv" "Downstream-Depths.csv")       ; Depth lookup table fo
set velocity-file-names         (list "Upstream-Vels.csv" "Middle-Vels.csv" "Downstream-Vels.csv")             ; Velocity lookup table

;; Reach-scale habitat parameters
set reach-drift-concs           (list 5.0E-10  5.0E-10  5.0E-10) ; Concentration of drift food, g/cm3
set reach-search-prods          (list 1.0E-5   1.0E-5   1.0E-5) ; Production of search food, g/cm2/d
set reach-shelter-speed-fracs   (list 0.3      0.3      0.3) ; Fraction by which velocity shelter reduces swimming speed
set reach-prey-energy-densities (list 2500     2500     2500) ; Energy content of fish prey, J/g
set reach-drift-regen-distances (list 1000     1000     1000) ; Distance over which drift is regenerated (cm)
set reach-shadings              (list 0.8      0.8      0.8) ; Multiplier for shading: fraction of sunlight *not* blocked by shading
set reach-fish-pred-mins        (list 0.9      0.9      0.9) ; Minimum probability of surviving fish predation
set reach-terr-pred-mins        (list 0.9      0.9      0.9) ; Minimum probability of surviving terrestrial predation
set reach-light-turbid-coefs    (list 0.0017   0.0017   0.0017) ; Multiplier in relation between light attentuation coefficient and tu
set reach-light-turbid-consts   (list 0.0      0.0      0.0) ; Constant in relation between light attentuation coefficient and turbidi
set reach-max-spawn-flows       (list 40       40       40) ; Maximum flow (m3/s) at which trout can spawn -- Grey River estimate
set reach-shear-As              (list 0.0      0.0      0.0) ; Multiplier (s/m3) in the relation between flow and shear stress, for re
set reach-shear-Bs              (list 1.0      1.0      1.0) ; Exponent in the relation between flow and shear stress, for redd scour
```

Figure 51. Example parameter file, part 3: Reach parameters.

14.3 Trout and redd parameters

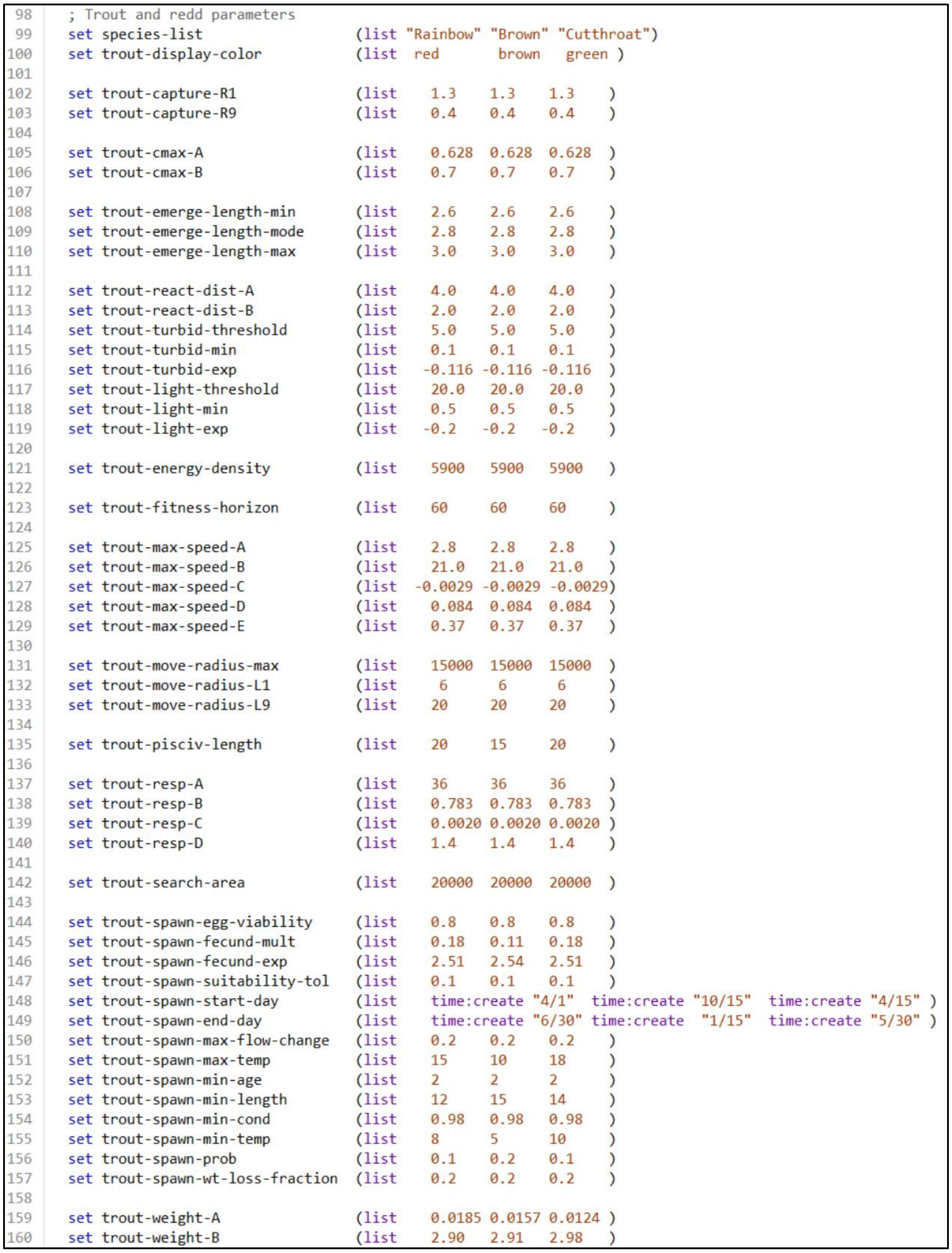
The last and biggest part of the parameter file provides values for trout and redd parameters. These parameters are numerous and thoroughly defined in the model description (Part II of this document) and we do not duplicate that description here. Instead, we provide Figure 52-Figure 57, which illustrate what this part of the parameter file looks like when edited in NetLogo. These figures also provide typical values of the parameters.

In Part IV, we provide guidance on which trout and redd parameters deserve re-evaluation for applications to new sites and species, and on how to develop new values. In general, we discourage replacing the standard parameter values with others taken from the literature without careful analysis and evaluation: what may appear to be differences among species or populations often really result from differences in study methods or in characteristics of the particular fish or habitat studied. This caution is especially relevant for bioenergetics parameters and parameters representing effects of temperature on trout and redd survival.

InSTREAM 7 is designed so that only the parameter file must be modified to change which species and how many species are simulated. This design is why each trout and redd parameter is entered not as a single value but as a list with one value for each species. The first trout parameters are ***species-list*** and ***trout-display-color*** (at the top of Figure 52). The value of ***species-list*** is a list of the names of the species in the simulation. All remaining trout and redd parameters must be lists of values for the same species, in the same order. For example, the second statement in Figure 52 tells inSTREAM that Rainbow Trout are displayed with the color red, that Brown Trout are displayed as brown, and Cutthroat Trout are displayed as green.

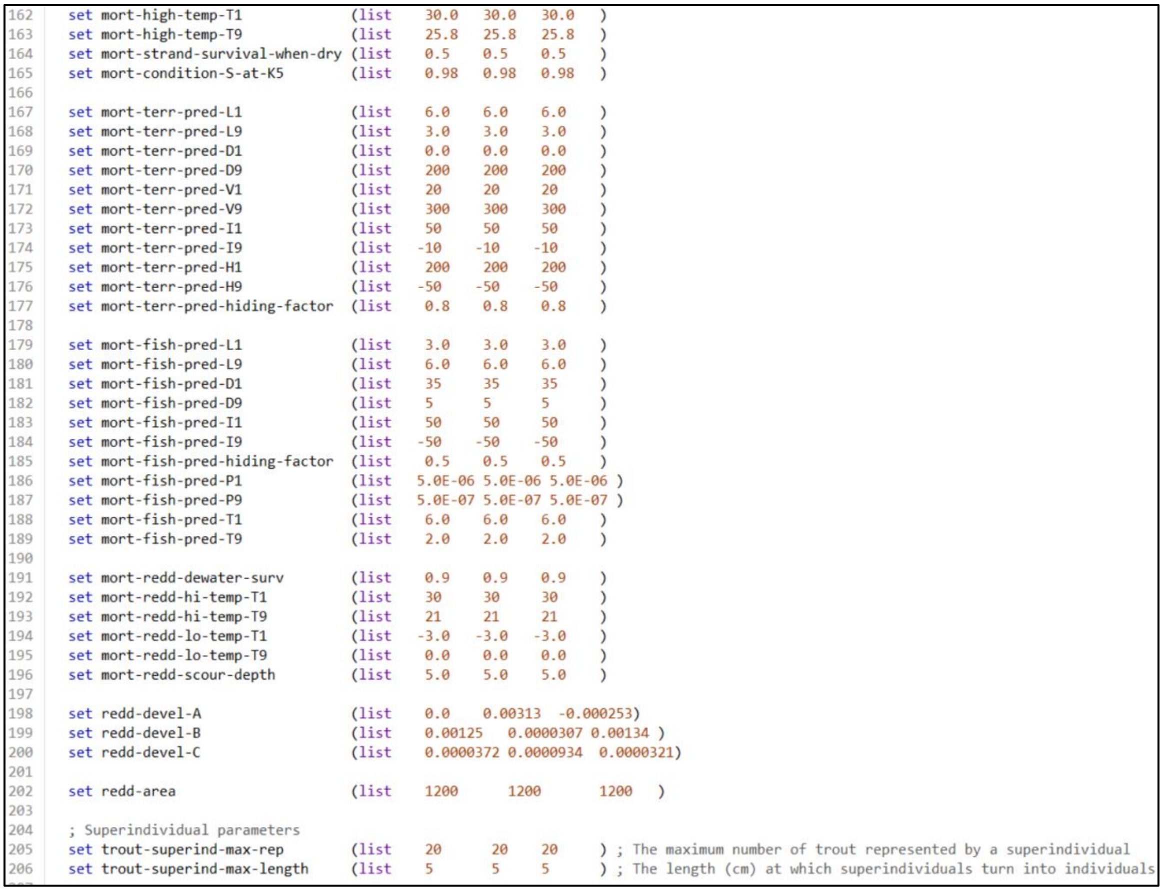
To add, remove, or change species, one must edit the values of ***species-list*** and ***trout-display-color*** to provide the desired species names and display colors, and then edit all the `set` statements for trout parameters to provide one value per species, in the species order defined by ***species-list***. These values must be inside the `(list ... )` statements as illustrated in the figures. Spacing and number formats are generally unimportant; values must be separated by one or more spaces or tabs.

Two parameters, for the dates defining the spawning season, also require the `time:create` command seen in Figure 52; it converts these dates from text to inSTREAM's date code. Only the dates, in M/d format, should be changed.



```
; Trout and redd parameters
set species-list                 (list "Rainbow" "Brown" "Cutthroat")
set trout-display-color          (list  red       brown   green )

set trout-capture-R1             (list    1.3    1.3    1.3     )
set trout-capture-R9             (list    0.4    0.4    0.4     )

set trout-cmax-A                 (list    0.628  0.628  0.628   )
set trout-cmax-B                 (list    0.7    0.7    0.7     )

set trout-emerge-length-min      (list    2.6    2.6    2.6     )
set trout-emerge-length-mode     (list    2.8    2.8    2.8     )
set trout-emerge-length-max      (list    3.0    3.0    3.0     )

set trout-react-dist-A           (list    4.0    4.0    4.0     )
set trout-react-dist-B           (list    2.0    2.0    2.0     )
set trout-turbid-threshold       (list    5.0    5.0    5.0     )
set trout-turbid-min             (list    0.1    0.1    0.1     )
set trout-turbid-exp             (list   -0.116 -0.116 -0.116   )
set trout-light-threshold        (list    20.0   20.0   20.0    )
set trout-light-min              (list    0.5    0.5    0.5     )
set trout-light-exp              (list   -0.2   -0.2   -0.2     )

set trout-energy-density         (list    5900   5900   5900    )

set trout-fitness-horizon        (list    60     60     60      )

set trout-max-speed-A            (list    2.8    2.8    2.8     )
set trout-max-speed-B            (list    21.0   21.0   21.0    )
set trout-max-speed-C            (list  -0.0029 -0.0029 -0.0029)
set trout-max-speed-D            (list    0.084  0.084  0.084   )
set trout-max-speed-E            (list    0.37   0.37   0.37    )

set trout-move-radius-max        (list    15000  15000  15000   )
set trout-move-radius-L1         (list     6      6      6      )
set trout-move-radius-L9         (list    20     20     20      )

set trout-pisciv-length          (list    20     15     20      )

set trout-resp-A                 (list    36     36     36      )
set trout-resp-B                 (list    0.783  0.783  0.783   )
set trout-resp-C                 (list    0.0020 0.0020 0.0020  )
set trout-resp-D                 (list    1.4    1.4    1.4     )

set trout-search-area            (list    20000  20000  20000   )

set trout-spawn-egg-viability    (list    0.8    0.8    0.8     )
set trout-spawn-fecund-mult      (list    0.18   0.11   0.18    )
set trout-spawn-fecund-exp       (list    2.51   2.54   2.51    )
set trout-spawn-suitability-tol  (list    0.1    0.1    0.1     )
set trout-spawn-start-day        (list    time:create "4/1"  time:create "10/15"  time:create "4/15" )
set trout-spawn-end-day          (list    time:create "6/30" time:create  "1/15"  time:create "5/30" )
set trout-spawn-max-flow-change  (list    0.2    0.2    0.2     )
set trout-spawn-max-temp         (list    15     10     18      )
set trout-spawn-min-age          (list    2      2      2       )
set trout-spawn-min-length       (list    12     15     14      )
set trout-spawn-min-cond         (list    0.98   0.98   0.98    )
set trout-spawn-min-temp         (list    8      5      10      )
set trout-spawn-prob             (list    0.1    0.2    0.1     )
set trout-spawn-wt-loss-fraction (list    0.2    0.2    0.2     )

set trout-weight-A               (list    0.0185 0.0157 0.0124  )
set trout-weight-B               (list    2.90   2.91   2.98    )
```

Figure 52. Example parameter file, part 4: Trout parameters (a)—species names, feeding and growth, and spawning.



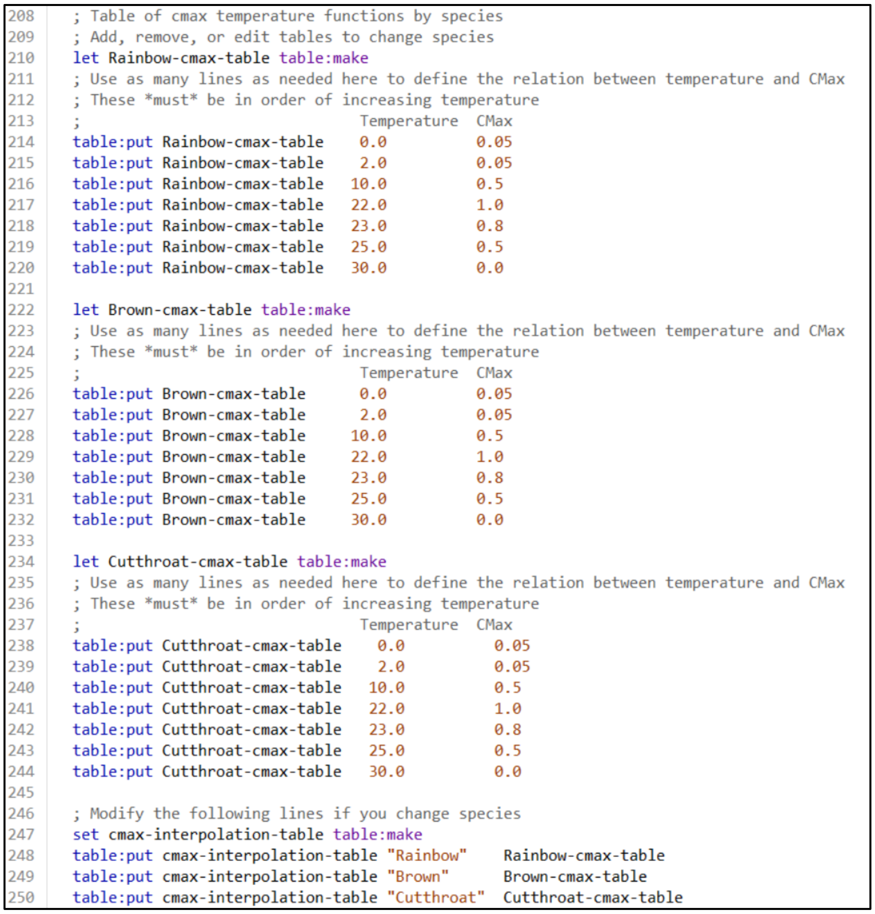
```
set mort-high-temp-T1            (list   30.0   30.0   30.0   )
set mort-high-temp-T9            (list   25.8   25.8   25.8   )
set mort-strand-survival-when-dry (list  0.5    0.5    0.5    )
set mort-condition-S-at-K5       (list   0.98   0.98   0.98   )

set mort-terr-pred-L1            (list   6.0    6.0    6.0    )
set mort-terr-pred-L9            (list   3.0    3.0    3.0    )
set mort-terr-pred-D1            (list   0.0    0.0    0.0    )
set mort-terr-pred-D9            (list   200    200    200    )
set mort-terr-pred-V1            (list   20     20     20     )
set mort-terr-pred-V9            (list   300    300    300    )
set mort-terr-pred-I1            (list   50     50     50     )
set mort-terr-pred-I9            (list  -10    -10    -10     )
set mort-terr-pred-H1            (list   200    200    200    )
set mort-terr-pred-H9            (list  -50    -50    -50     )
set mort-terr-pred-hiding-factor (list   0.8    0.8    0.8    )

set mort-fish-pred-L1            (list   3.0    3.0    3.0    )
set mort-fish-pred-L9            (list   6.0    6.0    6.0    )
set mort-fish-pred-D1            (list   35     35     35     )
set mort-fish-pred-D9            (list   5      5      5      )
set mort-fish-pred-I1            (list   50     50     50     )
set mort-fish-pred-I9            (list  -50    -50    -50     )
set mort-fish-pred-hiding-factor (list   0.5    0.5    0.5    )
set mort-fish-pred-P1            (list  5.0E-06 5.0E-06 5.0E-06 )
set mort-fish-pred-P9            (list  5.0E-07 5.0E-07 5.0E-07 )
set mort-fish-pred-T1            (list   6.0    6.0    6.0    )
set mort-fish-pred-T9            (list   2.0    2.0    2.0    )

set mort-redd-dewater-surv       (list   0.9    0.9    0.9    )
set mort-redd-hi-temp-T1         (list   30     30     30     )
set mort-redd-hi-temp-T9         (list   21     21     21     )
set mort-redd-lo-temp-T1         (list  -3.0   -3.0   -3.0    )
set mort-redd-lo-temp-T9         (list   0.0    0.0    0.0    )
set mort-redd-scour-depth        (list   5.0    5.0    5.0    )

set redd-devel-A                 (list   0.0    0.00313  -0.000253)
set redd-devel-B                 (list   0.00125   0.0000307 0.00134 )
set redd-devel-C                 (list   0.0000372 0.0000934  0.0000321)

set redd-area                    (list   1200      1200       1200   )

; Superindividual parameters
set trout-superind-max-rep       (list   20      20     20     ) ; The maximum number of trout represented by a superindividual
set trout-superind-max-length    (list   5       5      5      ) ; The length (cm) at which superindividuals turn into individuals
```

Figure 53. Example parameter file, part 5: Trout parameters (b)—survival, redds, and superindividual size.

InSTREAM uses three trout parameters that are each actually a table of values that define a nonlinear function. These parameters represent the effect of temperature on the maximum food consumption rate (*Cmax*; Sect. 9.24), and how spawning habitat suitability varies with cell depth and velocity (Sect. 9.29). These tables are entered in the parameter file as actual tables of values, with one table per species. (The code in the parameter file, which you need not understand, converts the values into an internal table format and stores a copy for each species.) These tables consist of several rows, each providing an X-Y pair for the function. For example, in Figure 54 the *Cmax* function for each species consists of 7 such rows, defining how *Cmax* changes as temperature increases from 0.0 to 30.0°C.

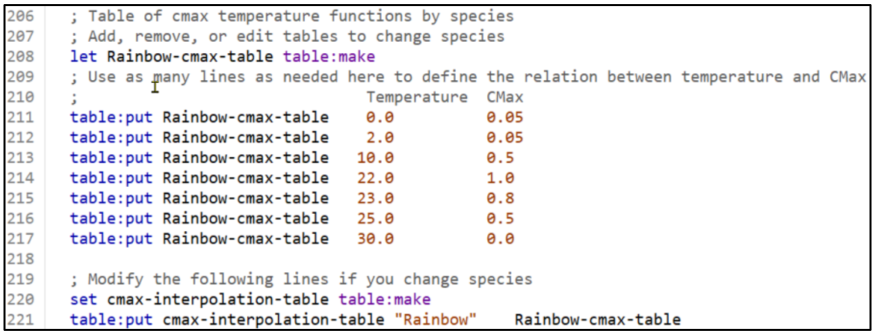
These tables are easily modified by editing the data values in the `table:put` statements. Rows can be added or deleted to modify the number of points in these functions. However, the rows defining each table *must be in increasing order of their X* (temperature, depth, velocity) values, and the first X value must be 0.0 (because water temperature, depth, and velocity cannot be less than zero). Different species can have different numbers of rows in their tables. The species names in the final `table:put` statements must exactly match those defined in the ***species-list*** parameter.

Changing the number of species requires, first, adding (by copying and modifying) or deleting tables so that one is provided for each species. Second, the statements that put each species table into a master table (the last lines in Figure 54) must be edited. For example, Figure 55 shows what the C_{Max} function parameters look like when converted from the three-species application of Figure 54 to a single-species model.



```
; Table of cmax temperature functions by species
; Add, remove, or edit tables to change species
let Rainbow-cmax-table table:make
; Use as many lines as needed here to define the relation between temperature and CMax
; These *must* be in order of increasing temperature
;                               Temperature  CMax
table:put Rainbow-cmax-table    0.0          0.05
table:put Rainbow-cmax-table    2.0          0.05
table:put Rainbow-cmax-table   10.0          0.5
table:put Rainbow-cmax-table   22.0          1.0
table:put Rainbow-cmax-table   23.0          0.8
table:put Rainbow-cmax-table   25.0          0.5
table:put Rainbow-cmax-table   30.0          0.0

let Brown-cmax-table table:make
; Use as many lines as needed here to define the relation between temperature and CMax
; These *must* be in order of increasing temperature
;                               Temperature  CMax
table:put Brown-cmax-table      0.0          0.05
table:put Brown-cmax-table      2.0          0.05
table:put Brown-cmax-table     10.0          0.5
table:put Brown-cmax-table     22.0          1.0
table:put Brown-cmax-table     23.0          0.8
table:put Brown-cmax-table     25.0          0.5
table:put Brown-cmax-table     30.0          0.0

let Cutthroat-cmax-table table:make
; Use as many lines as needed here to define the relation between temperature and CMax
; These *must* be in order of increasing temperature
;                               Temperature  CMax
table:put Cutthroat-cmax-table    0.0          0.05
table:put Cutthroat-cmax-table    2.0          0.05
table:put Cutthroat-cmax-table   10.0          0.5
table:put Cutthroat-cmax-table   22.0          1.0
table:put Cutthroat-cmax-table   23.0          0.8
table:put Cutthroat-cmax-table   25.0          0.5
table:put Cutthroat-cmax-table   30.0          0.0

; Modify the following lines if you change species
set cmax-interpolation-table table:make
table:put cmax-interpolation-table "Rainbow"    Rainbow-cmax-table
table:put cmax-interpolation-table "Brown"      Brown-cmax-table
table:put cmax-interpolation-table "Cutthroat"  Cutthroat-cmax-table
```

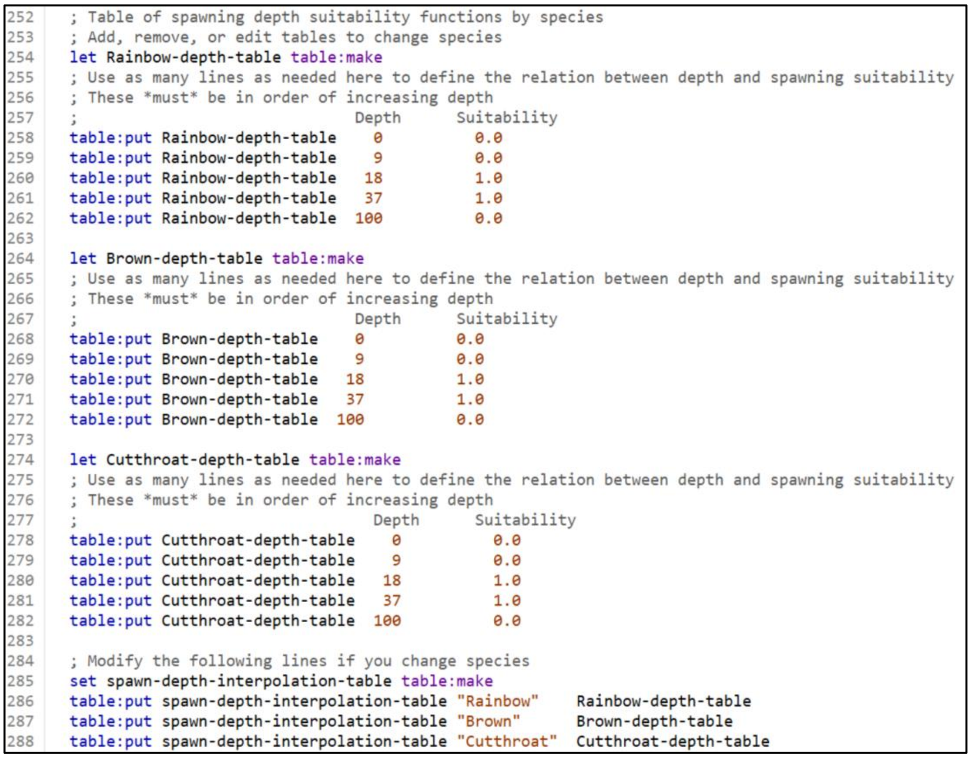
Figure 54. Example parameter file, part 6: Trout maximum consumption interpolation tables for a three-species application.



```
; Table of cmax temperature functions by species
; Add, remove, or edit tables to change species
let Rainbow-cmax-table table:make
; Use as many lines as needed here to define the relation between temperature and CMax
;                               Temperature  CMax
table:put Rainbow-cmax-table    0.0          0.05
table:put Rainbow-cmax-table    2.0          0.05
table:put Rainbow-cmax-table   10.0          0.5
table:put Rainbow-cmax-table   22.0          1.0
table:put Rainbow-cmax-table   23.0          0.8
table:put Rainbow-cmax-table   25.0          0.5
table:put Rainbow-cmax-table   30.0          0.0

; Modify the following lines if you change species
set cmax-interpolation-table table:make
table:put cmax-interpolation-table "Rainbow"    Rainbow-cmax-table
```

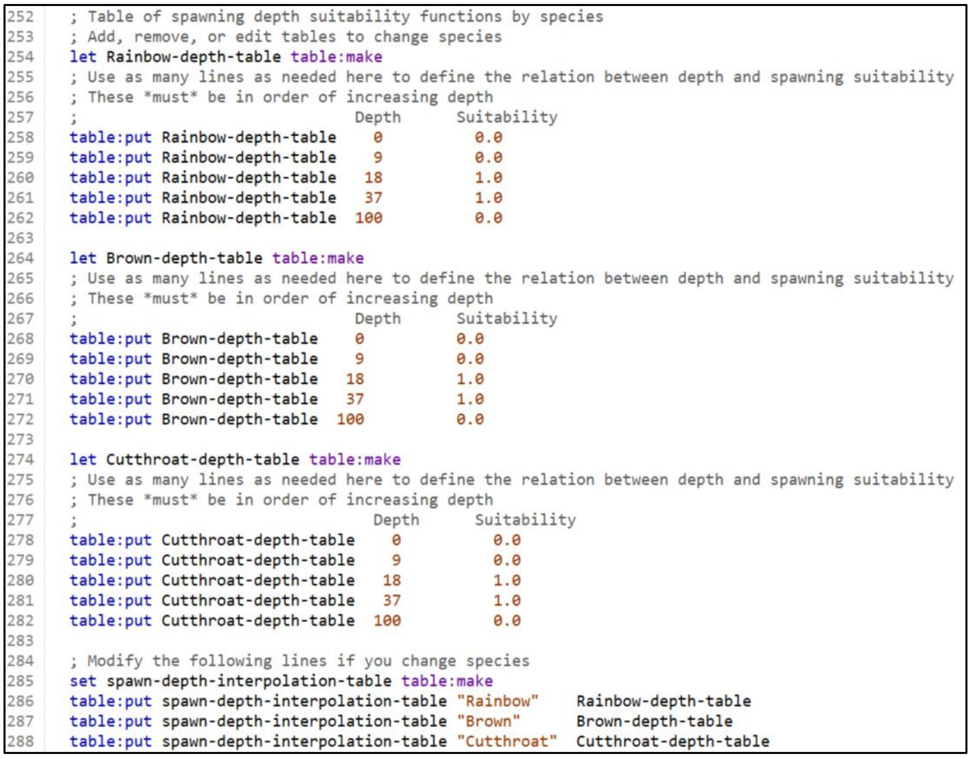
Figure 55. Trout maximum consumption interpolation table for a one-species application.

The statements providing the spawning depth and velocity suitability functions are illustrated in Figure 56 and Figure 57. These functions can be modified exactly as the Cmax table can, making sure that depth and velocity always start at 0.0 and increase from row to row.



```
; Table of spawning depth suitability functions by species
; Add, remove, or edit tables to change species
let Rainbow-depth-table table:make
; Use as many lines as needed here to define the relation between depth and spawning suitability
; These *must* be in order of increasing depth
;                               Depth      Suitability
table:put Rainbow-depth-table    0           0.0
table:put Rainbow-depth-table    9           0.0
table:put Rainbow-depth-table   18           1.0
table:put Rainbow-depth-table   37           1.0
table:put Rainbow-depth-table  100           0.0

let Brown-depth-table table:make
; Use as many lines as needed here to define the relation between depth and spawning suitability
; These *must* be in order of increasing depth
;                               Depth      Suitability
table:put Brown-depth-table    0          0.0
table:put Brown-depth-table    9          0.0
table:put Brown-depth-table   18          1.0
table:put Brown-depth-table   37          1.0
table:put Brown-depth-table  100          0.0

let Cutthroat-depth-table table:make
; Use as many lines as needed here to define the relation between depth and spawning suitability
; These *must* be in order of increasing depth
;                                 Depth      Suitability
table:put Cutthroat-depth-table    0           0.0
table:put Cutthroat-depth-table    9           0.0
table:put Cutthroat-depth-table   18           1.0
table:put Cutthroat-depth-table   37           1.0
table:put Cutthroat-depth-table  100           0.0

; Modify the following lines if you change species
set spawn-depth-interpolation-table table:make
table:put spawn-depth-interpolation-table "Rainbow"    Rainbow-depth-table
table:put spawn-depth-interpolation-table "Brown"      Brown-depth-table
table:put spawn-depth-interpolation-table "Cutthroat"  Cutthroat-depth-table
```

Figure 56. Example parameter file, part 7: Trout spawning depth suitability interpolation tables.



```
; Table of spawning depth suitability functions by species
; Add, remove, or edit tables to change species
let Rainbow-depth-table table:make
; Use as many lines as needed here to define the relation between depth and spawning suitability
; These *must* be in order of increasing depth
;                               Depth      Suitability
table:put Rainbow-depth-table    0          0.0
table:put Rainbow-depth-table    9          0.0
table:put Rainbow-depth-table   18          1.0
table:put Rainbow-depth-table   37          1.0
table:put Rainbow-depth-table  100          0.0

let Brown-depth-table table:make
; Use as many lines as needed here to define the relation between depth and spawning suitability
; These *must* be in order of increasing depth
;                               Depth      Suitability
table:put Brown-depth-table    0          0.0
table:put Brown-depth-table    9          0.0
table:put Brown-depth-table   18          1.0
table:put Brown-depth-table   37          1.0
table:put Brown-depth-table  100          0.0

let Cutthroat-depth-table table:make
; Use as many lines as needed here to define the relation between depth and spawning suitability
; These *must* be in order of increasing depth
;                                 Depth      Suitability
table:put Cutthroat-depth-table    0          0.0
table:put Cutthroat-depth-table    9          0.0
table:put Cutthroat-depth-table   18          1.0
table:put Cutthroat-depth-table   37          1.0
table:put Cutthroat-depth-table  100          0.0

; Modify the following lines if you change species
set spawn-depth-interpolation-table table:make
table:put spawn-depth-interpolation-table "Rainbow"    Rainbow-depth-table
table:put spawn-depth-interpolation-table "Brown"      Brown-depth-table
table:put spawn-depth-interpolation-table "Cutthroat"  Cutthroat-depth-table
```

Figure 57. Example parameter file, part 8: Trout spawning velocity suitability interpolation tables.

15 Simulation Experiments with BehaviorSpace

Using models like inSTREAM means designing and executing simulation experiments: sets of model runs designed to answer specific questions. In typical applications, simulation experiments are used to calibrate the model, to examine the sensitivity of its results to various inputs, and to address management questions such as which alternative instream flow scenario would best sustain a trout population and why.

NetLogo includes a tool for automating the execution of simulation experiments, called BehaviorSpace. After the user makes a few minor edits to the NetLogo file and fills out a short menu, BehaviorSpace automatically sets up the input for multiple model runs that each have different values of selected parameters, executes those runs in parallel on the computer's multiple processors, and compiles the results. Only a basic understanding of BehaviorSpace is needed to make it extremely useful; all but the most casual users of inSTREAM 7 will benefit from learning to use it.

15.1 Introduction to BehaviorSpace

The NetLogo User Manual's section on BehaviorSpace is the best place to begin learning about it. (Chapter 8 of Railsback and Grimm 2019 also provides an introduction to BehaviorSpace.) Here we assume that users have first read the User Manual section.

(The User Manual section on BehaviorSpace includes a subsection on Advanced Usage, including running experiments from the command line. This subsection is not relevant to typical users, but some may benefit from noting that the command line interface can be used to run large simulation experiments on computer clusters.)

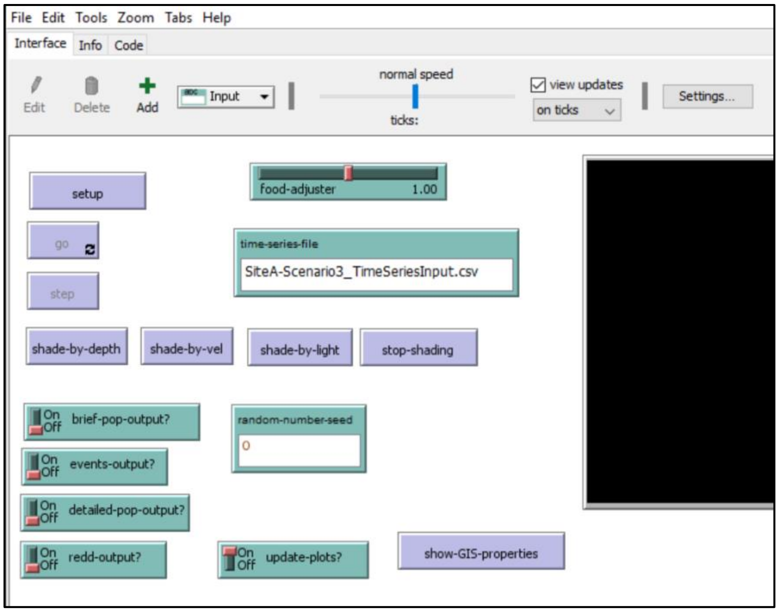
Briefly, using BehaviorSpace involves:

- Selecting one or a few parameters to vary. These can be any of the parameters in the parameter file: numbers, input file names, dates (e.g., the simulation start and end dates), and true-false switches.
- Selecting a set of values to use for each of these parameters.
- Deciding whether to produce replicates of each parameter combination: multiple model runs with the same parameter values but different random numbers.
- Specifying what output to record from each run (instead of, or in addition to, inSTREAM's standard output files).
- Letting BehaviorSpace automatically set up model runs for all combinations of the values of all selected parameters, with the selected number of replicates. For example, if we tell BehaviorSpace to use two values of the time-series input file name, and to vary the reach drift food concentration from 5.0 to 10.0 × 10^{-10} in steps of 1.0 × 10^{-10}, then BehaviorSpace will set up 12 model runs with all combinations of file name and drift concentration. If we also choose 5 replicates, we will have 60 model runs.
- BehaviorSpace then executes all the model runs and writes selected results from each run into its single output file.

To understand the following instructions for using BehaviorSpace, it is essential to understand exactly how it works. When BehaviorSpace starts each model run:

1. It first sets the values of the parameters that it varies.
2. Next, the Setup procedure is executed, as if we had clicked the Setup button. This first erases everything in the model, sets the values of all parameters to zero, then executes the code in the parameter file (Sect. 14) to re-set all parameter values. However, parameters that are defined on the Interface tab as sliders, switches, and inputs (Figure 58) are *not* altered by this setup step.
3. The model run starts and runs until finished.

This sequence means that BehaviorSpace can only control parameters that are defined on the Interface tab as shown in Figure 58. If instead we told BehaviorSpace to set the value of a parameter defined in the code and parameter file, the BehaviorSpace value would be erased and overwritten in the Setup step.



food-adjuster 1.00

time-series-file
SiteA-Scenario3_TimeSeriesInput.csv

random-number-seed
0

Figure 58. Example Interface showing parameters defined via a slider (`food-adjuster`), inputs (`time-series-file`; `random-number-seed`), and switches (for output files).

15.2 Experiment setup

We recommend the following sequence of steps for setting up and running a BehaviorSpace experiment. These steps may seem complicated to beginners, but we believe mastering them is not difficult and they can save a great deal of time.

Each BehaviorSpace experiment should be treated as a separate inSTREAM project: put it in its own directory with its own copy of the NetLogo file, the parameter file, and the input files. The NetLogo and parameter files will be modified for the experiment.

Step 1: Design the experiment by selecting the parameters to vary and their values. Common experiment designs are discussed in Sect. 24 and 18. Here we use two examples: a calibration experiment that varies both food availability and predation risk, and a contrast of alternative flow and temperature scenarios.

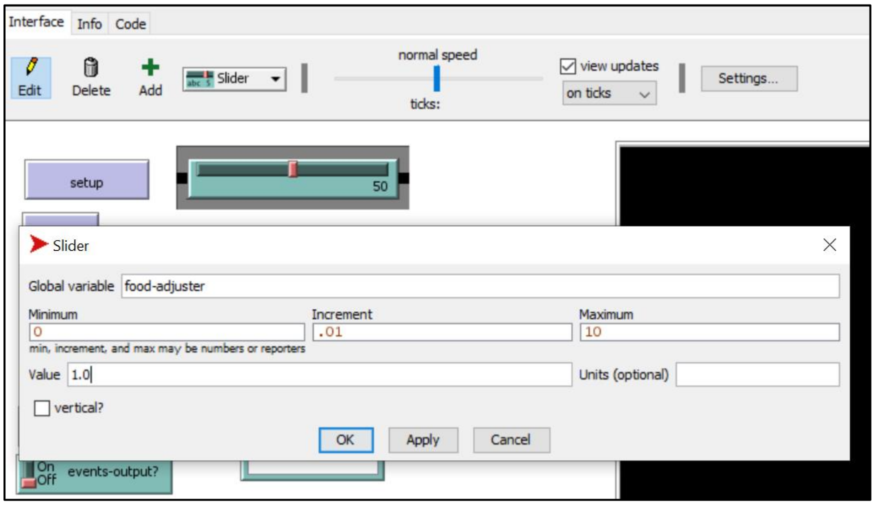
For the calibration experiment, we want to simulate 5 food availability scenarios that use drift food concentrations and search food production rates 25%, 50%, 100%, 150%, and 200% of the pre-calibration values. We also vary the parameter that controls survival of terrestrial predation (***reach-terr-pred-min***) from 0.95 to 0.98 in steps of 0.01. BehaviorSpace will therefore run the model 20 times, once for each combination of these parameter values.

For the experiment contrasting alternative flow and temperature scenarios, we use three different time series input files for the one reach we are modeling: `scen1-time-series-inputs.csv`, `scen2-time-series-inputs.csv`, and `scen3-time-series-inputs.csv`. We will execute five replicates of these scenarios, for 15 model runs.

Step 2: Define new parameter control variables on the Interface. We need to define new variables, just for the experiment we are running, that link BehaviorSpace to the parameters we want to control. BehaviorSpace will set the values of these new variables, and then the setup code in our parameter file will be modified to use the new variables to set the parameter values. We can do this in several ways.

For the calibration experiment, we want to vary two food parameters, ***reach-drift-conc*** and ***reach-search-prod***, by multiplying both by a fraction (from 25% to 200%) of their standard value. We can do this by defining a new variable ***food-adjuster*** that we then multiply the drift concentration and search food production by. This new variable is defined by creating a slider on the Interface (Figure 59). The NetLogo User Manual's Interface Tab Guide provides instructions for creating variables on the Interface.

(You can make up any names for control variables added to the Interface. If NetLogo displays a yellow error statement saying that there is already a global variable with the name you chose, simply go back to the Interface, right-click on the item you just added, and edit the variable name.)



Global variable: food-adjuster

Minimum: 0

Increment: .01

Maximum: 10

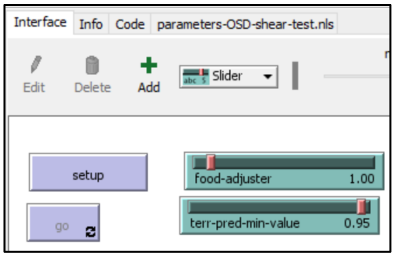
Value: 1.0

Units (optional):

□ vertical?

Figure 59. Creating a food-adjuster variable on the NetLogo Interface. When a slider is added to the Interface, this menu appears. Fill it out with the name of the variable being created, the range of its values, and its initial value.

We also need a slider to control the second parameter in the calibration experiment, ***reach-terr-pred-min***. For this parameter, we will let BehaviorSpace simply set its value instead of using an "adjuster" as we did for food. We create a new variable called ***terr-pred-min-value*** that ranges from 0.0 to 1.0 (Figure 60).

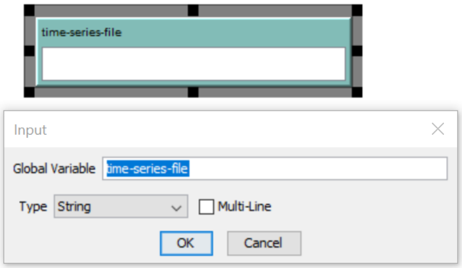


food-adjuster 1.00

terr-pred-min-value 0.95

Figure 60. The slider for controlling terrestrial predation survival.

Our second example experiment requires BehaviorSpace to control an input file name, which is a text (or "string") variable. Therefore, we need an "Input" item on the Interface that defines a character string. We add an input that defines a string variable we will call ***time-series-file*** (Figure 61).



Global Variable time-series-file

Type String

□ Multi-Line

Figure 61. Adding an "input" to define a string variable such as a file name.

Step 3: Edit the parameter file to use the new control variables to set parameter values. Now that we have created the control variables on the Interface, we need to edit the parameter file to use those variables to set the parameter values. This requires only some simple changes to the `set` statements in the parameter file.

For our example calibration experiment, we want to multiply the two food availability parameters by the value of the control variable ***food-adjuster***. We do this by changing these statements in the parameter file:

```
;; Reach-scale habitat parameters
set reach-drift-concs            (list 5.0E-10 ) ; Concentration of drift food, g/cm3
set reach-search-prods           (list 7.5E-7 ) ; Production of search food, g/cm2/d
```

to:

```
;; Reach-scale habitat parameters
set reach-drift-concs            (list (5.0E-10 * food-adjuster ) )
set reach-search-prods           (list (7.5E-7 * food-adjuster ) )
```

This example has only one reach, but the same approach is used with multiple reaches: multiply the standard parameter value by ***food-adjuster***, and put the multiplication for each reach within its own new set of parentheses; an example for two reaches is:

```
set reach-drift-concs (list (5.0E-10 * food-adjuster) (5.0E-10 * food-adjuster ))
```

To set the predation survival parameter to the Interface slider value of ***terr-pred-min-value***, simply change this parameter file statement:

```
  set reach-terr-pred-mins        (list 0.95)
```

to:

```
  set reach-terr-pred-mins        (list terr-pred-min-value)
```

For our second example experiment, we need to set the reach's time series input file to the string variable ***time-series-file*** that is in the Interface input. In the parameter file, simply change this:

```
  set time-series-input-files (list "ExampleSite_ObservedInput.csv")
```

to this:

```
  set time-series-input-files (list time-series-file)
```

(Note that quotation marks are not used around `time-series-file` because it is a variable name, not text.)

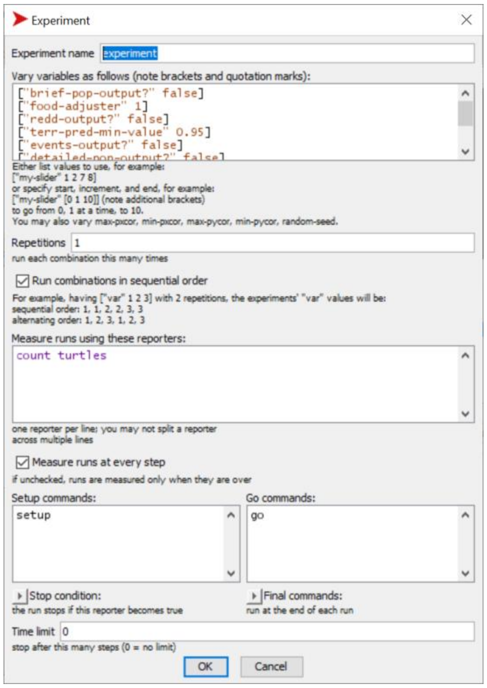
Step 4: Test and save the changes. After making these changes to the NetLogo file (adding control variables to the Interface) and parameter file, save both files. Because of some non-intuitive NetLogo behavior discussed in Sect. 14, after saving both the NetLogo and parameter files **you must also close them both and re-start NetLogo** to ensure that the changes are implemented in BehaviorSpace runs.

You should also test and debug the changes: simple mistakes such as misspelling variable names or forgetting parentheses are common. Test the changes by:

- In the Code tab or when editing the parameter file in NetLogo, click the "Check" button to run NetLogo's syntax checker. It will find most mistakes.
- Set up and run inSTREAM in the normal way, for at least a few time steps.

Step 5: Set up the BehaviorSpace experiment. From the NetLogo Tools menu, select BehaviorSpace. This opens a small dialog that lets you select an existing experiment or create a new one. (BehaviorSpace uses the word "experiment" for the settings that define one simulation experiment. These experiments are stored in the NetLogo file so they can later be edited, duplicated and edited into a new experiment, and executed. We normally distribute inSTREAM 7 with an example experiment saved in it.)

When you create a new experiment, BehaviorSpace opens a menu for you to edit (Figure 62). We recommend that you first give each experiment a new name; this name identifies the experiment for later re-use and also becomes part of the experiment's output file name.



Experiment

Experiment name: experiment

Vary variables as follows (note brackets and quotation marks):

```
["brief-pop-output?" false]
["food-adjuster" 1]
["redd-output?" false]
["terr-pred-min-value" 0.95]
["events-output?" false]
["detailed-pop-output?" false]
```

Either list values to use, for example:
["my-slider" 1 2 7 8]
or specify start, increment, and end, for example:
["my-slider" [0 1 10]] (note additional brackets)
to go from 0, 1 at a time, to 10.
You may also vary max-pxcor, min-pxcor, max-pycor, min-pycor, random-seed.

Repetitions: 1
run each combination this many times

☑ Run combinations in sequential order
For example, having ["var" 1 2 3] with 2 repetitions, the experiments' "var" values will be:
sequential order: 1, 1, 2, 2, 3, 3
alternating order: 1, 2, 3, 1, 2, 3

Measure runs using these reporters:

```
count turtles
```

one reporter per line; you may not split a reporter
across multiple lines

☑ Measure runs at every step
if unchecked, runs are measured only when they are over

Setup commands:

```
setup
```

Go commands:

```
go
```

Stop condition:
the run stops if this reporter becomes true

Final commands:
run at the end of each run

Time limit: 0
stop after this many steps (0 = no limit)

Figure 62. A new BehaviorSpace experiment.

In the “Vary variables...” box, BehaviorSpace automatically includes all the parameters defined on the Interface. For inSTREAM 7, these will include all the file output switches, the random number seed, plus the variables you put there to control an experiment. The file output switches should normally be set to `false`, but see Sect. 15.4 about using output files with BehaviorSpace.

You must edit the lines for your experiment control variables to tell BehaviorSpace what values to use in the experiment. For our example calibration experiment, we want ***food-adjuster*** to have several values from 0.25 to 2.0. Therefore (following the examples provided in the menu), we change the line:

```
["food-adjuster" 1]
```

to:

```
["food-adjuster" 0.25 0.5 1.0 1.5 2.0]
```

To give the survival parameter its values, we can use BehaviorSpace’s option to specify a starting value, an increment, and an ending value. We change this line:

```
["terr-pred-min-value" 0.95]
```

to this (note the additional set of brackets to tell NetLogo that we’re using start, increment, and end values):

```
["terr-pred-min-value" [0.95 0.01 0.98]]
```

For the scenario contrast experiment, we enter the three file names we want to use as values of the variable ***time-series-file***:

```
["time-series-file" "Scen1-time-series-inputs.csv" "Scen2-time-series-inputs.csv"
"Scen3-time-series-inputs.csv"]
```

You can set “Repetitions” in the experiment menu to a value higher than 1 if you want replicates of each parameter combination. For example, if “Repetitions” is set to 5, then BehaviorSpace will run five simulations with each combination of parameter values, with only the random numbers differing among the five runs. (If you use repetitions, the value of ***random-number-seed*** must be set to 0 on the Interface and in the experiment menu, if it appears there; otherwise each “repetition” will be identical.)

The final part of the experiment menu to change is the box labeled “Measure runs using these reporters”. In this box, you put variable names or code statements that define the outputs you want BehaviorSpace to produce in its output file. BehaviorSpace produces its own output file that includes results of all the model runs in the experiment, which is very convenient for analysis.

Using inSTREAM’s standard output files can eliminate the need to specify any outputs here in the experiment menu, and therefore the need to program global variables used only to report output in BehaviorSpace (explained below). We describe both options—using standard output files (in Sect. 15.4) and (here and in Sect. 16.6) using custom BehaviorSpace output variables.

BehaviorSpace’s output file automatically includes the values of all the parameters in the “Vary variables...” box, so you do not need to include them in the “Measure runs...” box. New experiments contain only the text `count turtles` in this box, which is not useful for inSTREAM; delete it. Instead, we typically output the date and time, the light phase (dawn, day, etc.), whether it is a census time step, and the number and mean length of trout by age class:

```
formatted-sim-time
light-phase
is-census?
age-0-abund
age-1-abund
age-2+-abund
age-0-length
age-1-length
age-2+-length
```

Each of the above outputs is a “global” variable that inSTREAM updates every time step. At Sect. 16.6 we provide information on customizing these output variables, e.g., to break them out by species or habitat reach. However, using inSTREAM’s standard output files as described in Sect. 15.4 can provide more detailed output without the need for these custom output variables.

The output lines can also include code statements that calculate an output value. For example, this statement could be included in the “Measure runs...” box:

```
sum [cell-area] of cells with [depth > 20]
```

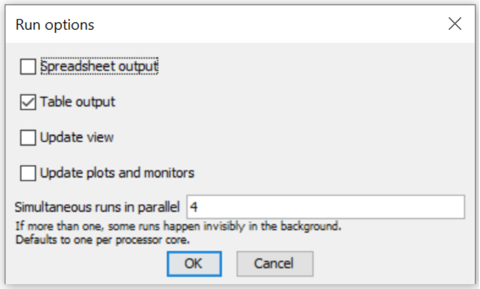
to output the total area of habitat with depth > 20 cm. Writing such custom output statements requires knowledge of NetLogo programming and how inSTREAM is programmed.

If you check the box “Measure runs at every step”, BehaviorSpace will output the state of the model every time step; if this check box is unchecked, you will get results only for the final time step, which is rarely useful for inSTREAM.

After making these edits to the experiment, click the “OK” button to exit the menu. It is smart to save your NetLogo file at this point, to save the experiment.

Step 6: Run the experiment. Now that the experiment is defined and saved, you can have BehaviorSpace execute all the model runs.

1. Open BehaviorSpace from the Tools menu.
2. Select the experiment you want to run, and click “Run”. This opens the Run options dialog (Figure 63).
3. From the options, *always* select “Table output” and *not* “Spreadsheet output”. (These options control how the output file is organized.)
4. The “Simultaneous runs in parallel” option displays how many processors (“threads”) your computer’s CPU has; by default BehaviorSpace will simultaneously run one model run per processor. (If your experiment has 24 model runs and your computer has 4 processors, BehaviorSpace will start 4 runs, and start another run every time an earlier one finishes, until all 24 are complete.) You can use this default to devote as many processors as possible to BehaviorSpace (you will still be able to do basic tasks on the computer but perhaps more slowly than normal), or you can let BehaviorSpace use all but 1 processor (set “Simultaneous runs in parallel” to one less than the default) to reserve a processor for other tasks.



Run options

☐ Spreadsheet output

☑ Table output

☐ Update view

☐ Update plots and monitors

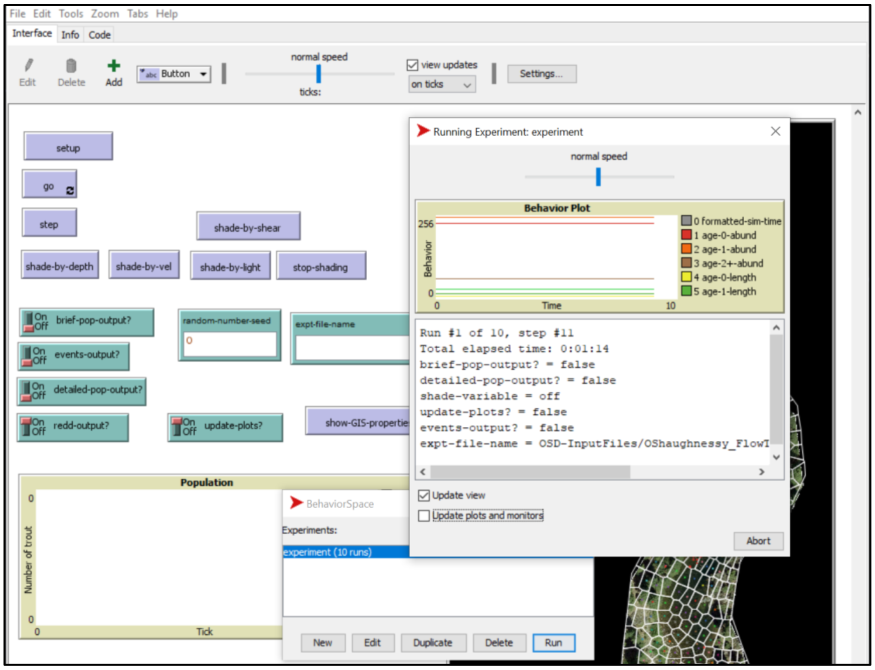
Simultaneous runs in parallel 4

If more than one, some runs happen invisibly in the background.
Defaults to one per processor core.

Figure 63. BehaviorSpace’s run options dialog.

5. Click OK. BehaviorSpace will then let you choose where it will write its output file. Normally, select the project directory.
6. BehaviorSpace now starts the model runs. While the runs execute, a “Running Experiments” dialog appears to display their progress (Figure 64; it takes a few seconds for this dialog to appear as the runs are set up). By checking the “Update view” and “Update plots and monitors” boxes, you can observe the progress of one selected run. These updates consume some computer resources so are normally turned off, though for inSTREAM their cost is small compared to that of running the model.
7. Watch the model interface for at least a few minutes as the first model runs are set up and started. Any problems with input files, parameter values, etc., are likely to show up as error dialogs that pop up in the background; they may be impossible to read behind the other dialogs. If you see anything that looks like an error dialog, click the “Abort” button to stop the experiment, then determine if something needs to be fixed.

8. The presence of the “Running Experiment” dialog indicates that BehaviorSpace is still running. (After the first model run is finished, it may appear that nothing is happening but the other model runs are executing as long as the dialog’s “Total elapsed time:” continues to increase.) When all runs are finished, this dialog will disappear.



```
Run #1 of 10, step #11
Total elapsed time: 0:01:14
brief-pop-output? = false
detailed-pop-output? = false
shade-variable = off
update-plots? = false
events-output? = false
expt-file-name = OSD-InputFiles/OShaughnessy_FlowT
```

☑ Update view

□ Update plots and monitors

Figure 64. The Running Experiment dialog that appears while BehaviorSpace is executing its model runs.

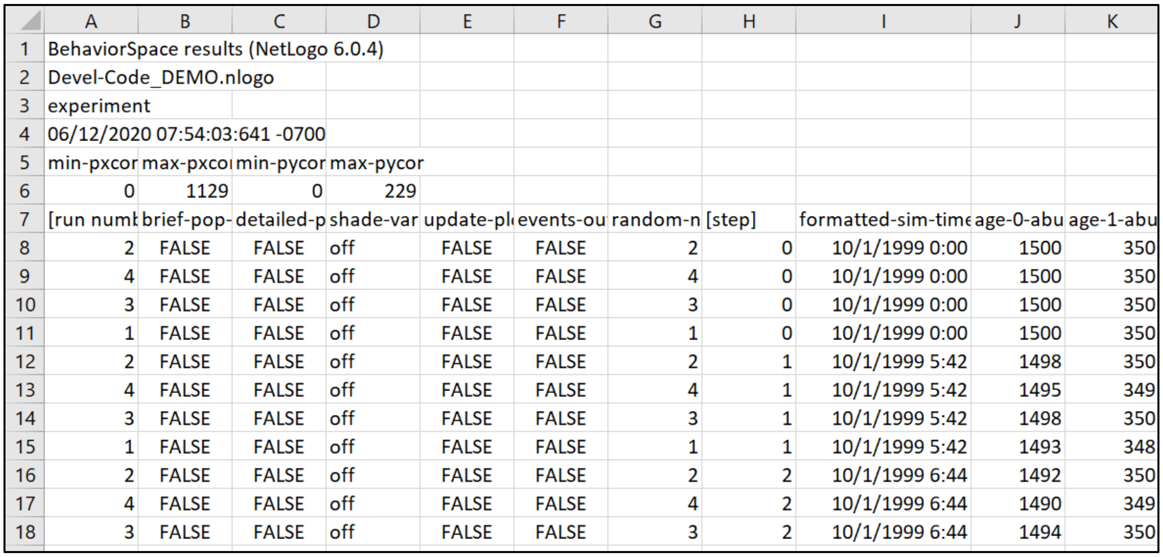
15.3 BehaviorSpace output

The BehaviorSpace output file provides results of all the experiment’s model runs in one file (Figure 65). Its header lines report the names of the NetLogo file and experiment that produced it, and when the experiment started. Then the columns report: (column A) the run number (the model runs executed by BehaviorSpace are numbered sequentially, starting with 1); (in this example, columns B-G) the value of all variables that were controlled in the “Vary variables...” box of the experiment menu (Figure 62); (column H) the time step (“tick” in NetLogo terminology) at which the output was written, and (all remaining columns) the values of all the outputs defined in the experiment menu. Note that the first output is written after the model is set up but before its first time step is executed (`[step]` has a value of 0).

You can see from Figure 65 that BehaviorSpace writes lines of output as each model run proceeds, so when multiple runs are executed in parallel on different processors their output is mixed together. You can sort the output by run number if you need to see each run’s results separately, but typically for analysis it is not necessary to sort the output file.

The output file is in CSV format and readily opened in spreadsheet or statistical software. Excel’s “PivotTable” tool is excellent for extracting and summarizing results from the file (without having to sort it). For example, in a PivotTable it is very easy to filter results and produce averages or annual time series from the time steps when ***is-census?*** is `true`.

BehaviorSpace "locks" its output file while an experiment is running, so you can open it in read-only mode to check progress before the experiment finishes, but do not try to save any changes to the file until the experiment finishes.



	A	B	C	D	E	F	G	H	I	J	K
1	BehaviorSpace results (NetLogo 6.0.4)										
2	Devel-Code_DEMO.nlogo										
3	experiment										
4	06/12/2020 07:54:03:641 -0700										
5	min-pxcor	max-pxco	min-pycor	max-pycor							
6	0	1129	0	229							
7	[run numb	brief-pop-	detailed-p	shade-var	update-pl	events-ou	random-n	[step]	formatted-sim-time	age-0-abu	age-1-abu
8	2	FALSE	FALSE	off	FALSE	FALSE	2	0	10/1/1999 0:00	1500	350
9	4	FALSE	FALSE	off	FALSE	FALSE	4	0	10/1/1999 0:00	1500	350
10	3	FALSE	FALSE	off	FALSE	FALSE	3	0	10/1/1999 0:00	1500	350
11	1	FALSE	FALSE	off	FALSE	FALSE	1	0	10/1/1999 0:00	1500	350
12	2	FALSE	FALSE	off	FALSE	FALSE	2	1	10/1/1999 5:42	1498	350
13	4	FALSE	FALSE	off	FALSE	FALSE	4	1	10/1/1999 5:42	1495	349
14	3	FALSE	FALSE	off	FALSE	FALSE	3	1	10/1/1999 5:42	1498	350
15	1	FALSE	FALSE	off	FALSE	FALSE	1	1	10/1/1999 5:42	1493	348
16	2	FALSE	FALSE	off	FALSE	FALSE	2	2	10/1/1999 6:44	1492	350
17	4	FALSE	FALSE	off	FALSE	FALSE	4	2	10/1/1999 6:44	1490	349
18	3	FALSE	FALSE	off	FALSE	FALSE	3	2	10/1/1999 6:44	1494	350

Figure 65. Example BehaviorSpace output file, as it appears in Excel.

15.4 BehaviorSpace and output files

Users can analyze BehaviorSpace experiments using inSTREAM's standard output files, in addition to or instead of the BehaviorSpace output file described in Sect. 15.3. We recommend the following procedure for doing so.

First, turn on the desired output files, as described in Sect. 12. When written by a BehaviorSpace experiment, separate output files are written for each model run. To indicate which model run produced which output file, the file names for main output files (Sect. 12.2) include the BehaviorSpace run number, and the output includes an additional column containing the run number.

Second, run the BehaviorSpace experiment. You must not open the inSTREAM output files with Excel while they are still being written to by BehaviorSpace; doing so will cause a NetLogo error when it tries to write to the opened file. However, (a) you can open the file during a run using software, such as the Notepad++ text editor, that does not lock the file when editing it; or (b) you can copy a file that BehaviorSpace is using and look at the copy.

Third, when the experiment finishes, combined all the output files of one type (e.g., the brief population output files) into one file that contains the output from all runs. The files can be combined by hand (e.g., by opening each in Excel and pasting their contents into one worksheet), using scripts in platforms such as R, or by using operating system commands. In Windows, you can:

1. Open the Windows Command Prompt (under the Windows System menu).
2. Navigate to the directory containing the output files, using the `cd` (change directory) command; e.g., `cd MyProject\InSTREAM\CalibrationExpt2`
3. Use the `copy` command to concatenate all the output files of one type into a single file; e.g., the command `copy BriefPopOut* BriefPop-AllRuns.csv` will concatenate all the brief population output files in the directory into the new file `BriefPop-AllRuns.csv`.

You can alternatively use the Windows Powershell, in which the command to combine files is:

```
get-content BriefPopOut* | set-content BriefPop-AllRuns.csv
```

In MacIntosh and Linux operating systems, open a terminal window, navigate to the output directory using `cd`, and combine files using the cat command: `cat BriefPopOut* > BriefPop-AllRuns.csv`

Now you can open and analyze the combined file, but you will encounter three potential problems. First, the combined file will contain the header lines from each separate output file. These extra header lines can be removed by sorting the file by, e.g., the BehaviorSpace run number.

The second problem is that the combined file can be very large, often too large to open in Excel. We provide a tool on the InSTREAM web site to reduce file size by keeping only the desired lines. This tool is a NetLogo program (`OutputFileFilterer.nlogo`) that is easily customized to define which output lines to keep, e.g., those for specific age classes or ranges of dates. The tool eliminates extra header lines at the same time.

The third problem is that InSTREAM's standard output files do not report the values of the parameters that were varied among the BehaviorSpace runs; they only report the run number. If you need the output file to include columns for these parameter values, you can use the BehaviorSpace output file (Sect. 15.3) as a lookup table that provides the parameter values for each run. You can, for example, open the BehaviorSpace output file in Excel, copy it into the workbook containing the other outputs, and use Excel's `vlookup` function to report parameter values associated with each run number.

15.5 Year shuffler replication

The year shuffler (sects. 8.2 and 14.1.3) is designed as an alternative way to replicate simulation experiments, so it is typically used within BehaviorSpace. However, using it correctly requires some care.

We use the year shuffler as an alternative to using the replicates option in the BehaviorSpace experiment setup (Sect. 15.2); instead of only changing the random numbers among model runs, the year shuffler lets us run a set of scenarios multiple times, each time with a different sequence of water years. There are two ways to do this, with different consequences.

The easiest (but usually not best) way to use year shuffling is to: (a) set the year shuffler's parameter values (Sect. 14.1.3), and set the parameter ***shuffle-years?*** to `true`, (b) set both ***random-seed*** and ***shuffle-rand-seed*** to `0`, and (c) set BehaviorSpace's replicates to how many year-shuffler replicates you want. This will execute each scenario multiple times, with input years shuffled each time. However, it is very important to understand that the sequence of water years will be randomized in each model run, so the different scenarios will be executed with different sets of shuffled water years. Consequently, some differences in results among scenarios will result from the different water year shuffles used for each scenario.

The second, more common, way to use the shuffler causes all scenarios to be executed using the same sequences of water years. We do this by using ***shuffle-rand-seed***: by giving each scenario the same set of shuffler random seeds, we assure that all scenarios experience the same set of time series inputs. Do this by:

- Creating a slider on the interface containing a variable named something like `shuffle-seed-control`,
- In the parameter file, change `set shuffle-rand-seed 0` to `set shuffle-rand-seed shuffle-seed-control`.
- In the BehaviorSpace experiment setup, add this to the variables to be controlled: `["shuffle-seed-control" 1 2 3 5 6]`

This will create, for each scenario, one year-shuffler replicate for each value given to `"shuffle-seed-control"`. (Note that `"shuffle-seed-control"` numbers must be integers, but their values are otherwise unconstrained.)

Keep in mind that the year shuffler automatically writes its own output file for each model run. These outputs show what actual years of data were used as shuffled input so you can understand and verify what the shuffler did.

16 Software Modification

Making inSTREAM's software easier for users to modify was a major motivation for programming it in NetLogo. Even though the model can be applied to new study sites and trout species with changes only to the parameter file, our experience with previous versions leads us to expect that many users will want to make small changes to the model code. Users are also likely to want to modify inSTREAM's graphical interface, especially by adding or changing plots. Here, we provide guidance for several kinds of change most likely to be needed. Regular users of inSTREAM 7 will likely find it worthwhile and easy to learn enough about NetLogo to understand, write, and test code and to modify and add graphics to the Interface.

This section includes many code examples. We remind readers that in NetLogo's programming language any text to the right of a semicolon is a comment ignored by the computer.

16.1 General procedures for modifying code

NetLogo provides a very powerful programming language with many built-in commands ("primitives") specifically for individual-based modeling, plus excellent error-checking tools. Unlike conventional programming languages, NetLogo does not require a separate compilation stage: we can edit the code and then run it immediately without having to convert code files into an executable program file.

NetLogo programs are divided into *procedures*: modular blocks of code that each implement a submodel or function. At the top of the Code tab in NetLogo, there is a button labeled

"Procedures"; clicking on it opens a list of procedures in the code, and clicking on a procedure's name takes you to it. There are roughly 90 procedures in the inSTREAM code. Each starts with comments describing what the procedure does and what kind of model entity (observer, trout, cell, ...) executes it.

In general, modifying the inSTREAM 7 code involves the following steps.

1. Decide exactly what you want to change and how, and thoroughly describe the change in a written document. If you want to make a substantial change in a submodel, program the new submodel independently (in Excel, R, etc.) and explore it thoroughly and work out any problems before trying to implement it in inSTREAM. Even for small changes (like the ones we illustrate in the following subsections), it will be important to know exactly what you want to do before changing the code.
2. Create a new project by copying an existing one. Never attempt to modify the model without keeping a copy of the unmodified code and input. Give the NetLogo file a new name, and keep notes in its Info tab documenting the changes.
3. Find the place(s) in the code needing modification. The following subsections tell you where to make common changes. For other changes, remember that the model description of Part II includes footnotes saying where each part of the model is coded.
4. Make the code changes. Programming almost always requires frequent use of NetLogo's Dictionary to find and understand useful primitives.
5. Check the revised code for errors by clicking the Code tab's "Check" button. This button runs a syntax checker that will find many kinds of mistakes.
6. Save the changed NetLogo file.
7. Make a test run. Mistakes often do not appear until the model is executed, and NetLogo is excellent (but not perfect) at telling you exactly what is wrong and where.
8. Test your change. Even though NetLogo's language is very easy to use, it does not by any means guarantee mistake-free code. In fact, it can be very easy to write code that looks like it does what you want when in fact it does not. *Always* test any change in the code immediately and thoroughly, instead of risking a mistake that costs a great deal of time. InSTREAM's debugging output files often provide enough information to test code changes. Additional test procedures and example code tests are available from the inSTREAM web site, and Chapter 6 of Railsback and Grimm (2019) provides numerous testing methods.

For changes that involve the number of trout (e.g., adding output related to trout abundance), be aware that the use of superindividuals (Sect. 4.1.4) complicates calculation of abundance. The NetLogo statement `count trout` (or: `count trout with [trout-species = "Rainbow"]`, e.g.) reports the current number of model trout, but some model trout may be superindividuals representing more than one fish. Therefore, the correct way to report the number of trout is: `sum [trout-superind-rep] of trout`. (or, e.g.: `sum [trout-superind-rep] of trout with [trout-species = "Rainbow"]`).

16.2 Modifying the population output files

InSTREAM 7's standard and secondary output files provide results in various levels of detail (Sect. 12). Often, though, users can simplify model analysis by adding variables to an output file, or by removing detail to better summarize results. These changes require code edits, but the code design makes this easy.

The main output files are defined in two procedures. The file headers (information at the top of the file and column labels) are defined in `build-output-file-named`, while the output lines are produced in `update-output`. Adding or removing variables from these files requires changes in both of these procedures.

Here are examples of the two kinds of changes you can make to the two main population output files: adding an output variable, and adding a breakout variable. (We address the events output files separately at Sect. 16.3. The redd output file is less likely to need modification and we do not address it here.)

Adding an output variable. First, let's modify the brief population output file to add the mean cell depth of trout. Without explaining the details, let us tell you that the code `mean [depth] of` calculates the mean depth occupied by a set of trout. (While the model description defines depth as the cell variable *cell-depth*, the code treats depth as a patch variable so it can be displayed as a color.) We start by adding depth as a labeled column in the brief population output file, modifying this code that creates the file in the procedure `build-output-file-named`:

```
to build-output-file-named [a-file-name]
  ; An observer procedure to initialize an output file.
  ; The parameter a-file-name is the global variable for the file name.

  ; Create the brief population output file.
  if a-file-name = "b-p-o-n" ; This is the value of the uninitialized file name
  [
    set brief-pop-outfile-name (word "BriefPopOut-" outfile-name-base)
    if file-exists? brief-pop-outfile-name [ file-delete brief-pop-outfile-name ]
    file-open brief-pop-outfile-name
    file-print (word "InSTREAM 7 brief population output file, Created " date-and-
time)
    file-print "End of time step,IsCensus?,Light phase,Reach,Species,Age class,
Count,Mean length,Mean weight,Mean condition,FractionDriftFeeding,
FractionSearchFeeding,FractionHiding"
    file-close
  ]
```

The column headers are written in the final `file-print` statement, which we change by adding mean depth as the final output variable:

```
    file-print "End of time step,IsCensus?,Light phase,Reach,Species,Age class,
Count,Mean length,Mean weight,Mean condition,FractionDriftFeeding,
FractionSearchFeeding,FractionHiding,Mean depth"
```

Then we modify this code in `update-output` that calculates the output each time step:

```
    ; Do the brief population output
    if brief-pop-output?
    [
      if brief-pop-outfile-name = "b-p-o-n" [ build-output-file-named brief-pop-
outfile-name ]
      file-open brief-pop-outfile-name

      foreach reach-names [ next-reach ->
        foreach species-list [ next-species ->
          foreach age-class-list [ next-age ->
           let last-age? (next-age = last age-class-list)  ; last-age? is true if..
           let age-label (word "Age-" next-age)
           let the-fish nobody
           ifelse last-age?
           [
             set age-label (word age-label "+")
             set the-fish trout with [ ([reach-name] of patches-reach = next-reach)
and (trout-species = next-species) and (trout-age >= next-age) ]
           ]
           [ set the-fish trout with [ ([reach-name] of patches-reach = next-reach)
and (trout-species = next-species) and (trout-age = next-age) ]
           ]

           ; These statements are to make output work when there are no fish...
           ; (because "mean" raises an error if there are no values).
           let the-count 0.0
           let the-length 0.0
           let the-weight 0.0
           let the-condition 0.0
           let the-frac-drift 0.0
           let the-frac-search 0.0
           let the-frac-hide 0.0

            if any? the-fish
            [
              set the-count (sum [trout-superind-rep] of the-fish)
              set the-length (mean [trout-length] of the-fish)
              set the-weight (mean [trout-weight] of the-fish)
              set the-condition (mean [trout-condition] of the-fish)
              set the-frac-drift (count the-fish with [ trout-activity = "drift" ] /
count the-fish)
              set the-frac-search (count the-fish with [ trout-activity = "search" ] /
count the-fish)
              set the-frac-hide (count the-fish with [ trout-activity = "hide" ] /
count the-fish)
            ]

              file-print csv:to-row (list formatted-sim-time is-this-a-census-date?
light-phase next-reach next-species age-label
                 the-count
                 the-length
                 the-weight
                 the-condition
                 the-frac-drift
                 the-frac-search
                 the-frac-hide
                ) ; End of output list
```

Clearly, this code breaks the trout population down into species and age class sets, then calculates summary statistics on each such set and writes them. To add the mean depth of each, we add it as the last of the `let` statements:

```
          let the-frac-hide 0.0
          let the-depth 0.0
```

and as the last of the final `set` statements:

```
             set the-frac-hide (count the-fish with [ trout-activity = "hide" ] /
count the-fish)
             set the-depth (mean [depth] of the-fish)
```

and to the end of the `file-print` statement:

```
                the-frac-hide
                the-depth
               ) ; End of output list
```

Adding a breakout variable. Now we show how to modify the same output file to break results out by an additional variable by which trout can be categorized. The code above breaks out the population summary (i.e., writes a separate line of output) by habitat reach, trout species, and trout age class. If we also want results separated by trout sex (the variable *trout-sex*, which has values of “male” and “female”), we need to add sex to the file headers and to the code that breaks out and calculates summary statistics. Because the code for age class is more complex, it will be easiest to add sex after species and before age class. In the procedure `build-output-file-named`, we add “Sex” as a column header, so the statement that writes column headers becomes:

```
    file-print "End of time step,IsCensus?,Light phase,Reach,Species,Sex,Age class,
Count,Mean length,Mean weight,Mean condition,FractionDriftFeeding,
FractionSearchFeeding,FractionHiding"
```

In the procedure `update-output`, we need to add a statement that defines a list of values for *trout-sex* and iterates through the list while defining the sets of trout to calculate statistics on. We change this code:

```
      foreach reach-names [ next-reach ->
        foreach species-list [ next-species ->
          foreach age-class-list [ next-age ->
```

to this:

```
      foreach reach-names [ next-reach ->
       foreach species-list [ next-species ->
        foreach (list “female” “male”) [ next-sex ->
         foreach age-class-list [ next-age ->
```

Then we need to modify the statements that identify the fish in each set to this:

```
           [
             set age-label (word age-label "+")
             set the-fish trout with [ ([reach-name] of patches-reach = next-reach)
and (trout-species = next-species) and (trout-age >= next-age) and (trout-sex = next-
sex) ]
           ]
           [ set the-fish trout with [ ([reach-name] of patches-reach = next-reach)
and (trout-species = next-species) and (trout-age = next-age) and (trout-sex = next-
sex) ]

           ]
```

We also must add sex to the breakout variables printed out:

```
              file-print csv:to-row (list formatted-sim-time is-this-a-census-date?
light-phase next-reach next-species next-sex age-label
```

Finally, we need to close the new loop created by the new `foreach` statement, at the end of the code for this output file:

```
                ) ; End of output list
           ] ; foreach age-class
          ] ; foreach trout-sex
        ] ; foreach species-list
      ] ; foreach reach-names
```

16.3 Events output

The two events output files simply report the state of a trout or redd when some event happens to it (Sect. 12.2). The code design makes it easy to report more kinds of events in these files.

Searching the code for “`save-event`” finds statements such as these:

```
      save-event "died of high temperature"
```

and

```
      save-event (word "lo-temp killed " eggs-died " eggs")
```

Each such statement creates a line of output in one of the events output files: the inSTREAM code determines whether a redd or a trout executed the statement, then writes its state variables and the text in the `save-events` statement to the appropriate output file. (The code `(word ...)` creates a text string by concatenating several strings and numbers so it says, in the above example, how many eggs died of low temperature mortality.)

Users can easily have these files report additional events by simply inserting the statement `save-event "text identifying the event type"` in code executed by a trout or a redd. For example, when a female trout spawns it uses this code to tell its male mate to record that it spawned this season and reduce its weight:

```
      ask max-one-of potential-mates [trout-length]
      [
        set trout-spawned-this-season? true
        set trout-weight trout-weight * (1.0 - (item trout-spp-index trout-spawn-wt-
loss-fraction))
      ] ; ask mate
```

We could make the male mate record its spawning event this way:

```
      ask max-one-of potential-mates [trout-length]
      [
        set trout-spawned-this-season? true
        set trout-weight trout-weight * (1.0 - (item trout-spp-index trout-spawn-wt-
loss-fraction))
        save-event "male spawned"
      ] ; ask mate
```

(We put the `save-event` statement *after* the statement causing spawning weight loss so that the events output will report the male’s length and weight after this weight loss.)

In adding events to be reported in this way, be very careful not to save events that happen extremely frequently. For example, saving events that trout do every time step (feeding, growing, surviving) would make the trout events output file extremely large and cumbersome.

16.4 Modifying secondary output files

Users may also need to modify inSTREAM’s secondary output files (Sect. 12.3). The individual fish and individual cell output files are coded like the main population output files: the file headers are written in the procedure `build-output-file-named` while the output is written in `update-output`. Therefore, the above guidance for modifying the population output files also applies to these files.

The individual cell output file by default only produces output for cells that are not dry (their depth is greater than zero). There are comments in the update-output code for this file that tell how to instead report output for all cells.

The "debug" output files are not produced by build-output-file-named and update-output. They are instead each written within the procedure that implements the submodel they report on. The code for a debug file can be found by searching the code (control-F) for the parameter that turns the file on and off. For example, to modify the debug output for redd survival, search the Code tab for its switch parameter debug-redd-survival?. This search will take you to the beginning of the procedure redd-survive, where you will find code to both create and write to the file. Adding a variable to these output files requires two steps. First is adding a column label to this code that creates the file and writes its headers:

```
  if not file-exists? debug-redd-survival-file-name
    [
      file-open debug-redd-survival-file-name
      file-print (word "InSTREAM 7 redd survival test output file, Created:," date-
and-time  ",Values are *daily* survival rates")
      file-type "redd,species,reach,temperature,flow,depth,is-flow-peak?,scour-param,"
      file-print "s-lo-temperature,s-hi-temperature,s-dewater,s-scour"
      file-close
    ]
```

(The command file-type writes part of a line, not including a line-end; file-print includes a line-end character.) The second step is adding the new variable to the list of those reported, in this statement:

```
    file-print csv:to-row (list
      who
      redd-species
      [reach-name] of patches-reach
      [reach-temperature] of patches-reach
      [reach-flow] of patches-reach
      depth
      ...
    )
```

16.5 Modifying graphical output

NetLogo makes it easy to add graphical output to the Interface tab. Built-in tools ("widgets") for observing models include line plots, histograms, "monitors" that each display the current value of one model variable or piece of code that calculates a value (e.g., sum [trout-superind-rep] of trout), and "outputs"—windows that the model can write to. InSTREAM 7 is normally distributed with an example line plot. NetLogo comes with thorough documentation of how to add and use its display widgets; Chapter 9 of Railsback and Grimm (2019) provides additional guidance, especially for histograms. We add the following guidance specific to inSTREAM.

NetLogo allows users to put the code that updates a widget directly in the widget. For example, when you add a plot to the Interface, NetLogo provides a menu with example code statements that tell the widget what to plot. We strongly discourage this and highly recommend (a) deleting all code from the display widgets and instead (b) adding code to update the widget to the existing display code in the procedure update-outputs, using the existing plot code as an example. (Railsback and Grimm 2019 explain the reasons for this recommendation.) If a plot does not look right, edit it on the Interface and look for any code that needs to be deleted.

We do not recommend using "monitor" widgets because they automatically update several times per second. Therefore, they can consume substantial computer resources, and there typically is no need for such rapid updates. Instead, add code to the `update-outputs` procedure that writes to an Output.

When writing output code that reports trout abundance, remember the caution about superindividuals in Sect. 16.1. Unfortunately, the use of superindividuals makes it difficult to produce accurate histograms for trout with length less than the parameter ***trout-superind-max-length***. NetLogo makes it very easy to produce histograms for trout variables by using statements such as `histogram [trout-length] of trout`. However, the histogram will reflect the number of trout entities without considering how many individuals each one represents. The only simple ways of dealing with this are to either not use superindividuals or simply be aware that histograms under-represent small trout by counting each superindividual as one trout.

It could be tempting to use the BehaviorSpace output variables (Sects. 15.3, 16.6) in plots and outputs, but that will not work without a small code change. Near the end of `update-outputs` is this statement:

```
  ; Update BehaviorSpace outputs, if this is a BehaviorSpace experiment
  if behaviorspace-run-number > 0 [ update-behaviorspace-outputs ]
```

To use the BehaviorSpace outputs in plots, delete that statement and instead add a statement at the beginning of the procedure:

```
 to update-output
  ; Observer procedure to update graphics and write file output
  update-behaviorspace-outputs
```

This change will update the BehaviorSpace outputs every time step, which, of course, requires additional computations.

16.6 Modifying BehaviorSpace outputs

In Sect. 15.2 we show how BehaviorSpace can output any kind of model result if the model code is programmed to calculate that result as a global variable. Therefore, when we want additional kinds of output from BehaviorSpace we typically need to add global variables and code to give them the values we want. Here we provide examples of doing this.

The BehaviorSpace setup example in Sect. 15.2 includes output of trout abundance and mean length for each of three age classes. We modify these outputs to get separate results for each species, for a model representing Rainbow and Brown Trout.

First we need to create global variables for all the results we want. All the global variables are defined near the top of the Code tab. In the long list of global variables is a group identified specifically as BehaviorSpace outputs:

```
  ;; BehaviorSpace outputs, updated in update-BehaviorSpace-outputs.
  age-0-abund       ; Abundance of all age 0 trout, regardless of species
  age-1-abund       ; Abundance of all age 1 trout
  age-2+-abund      ; Abundance of all trout age 2 and older
  age-0-length      ; Mean length of all age 0 trout, all species; blank if none
  age-1-length      ; Mean length of all age 1 trout, all species; blank if none
  age-2+-length     ; Mean length of all age 2 and older trout, all species;
```

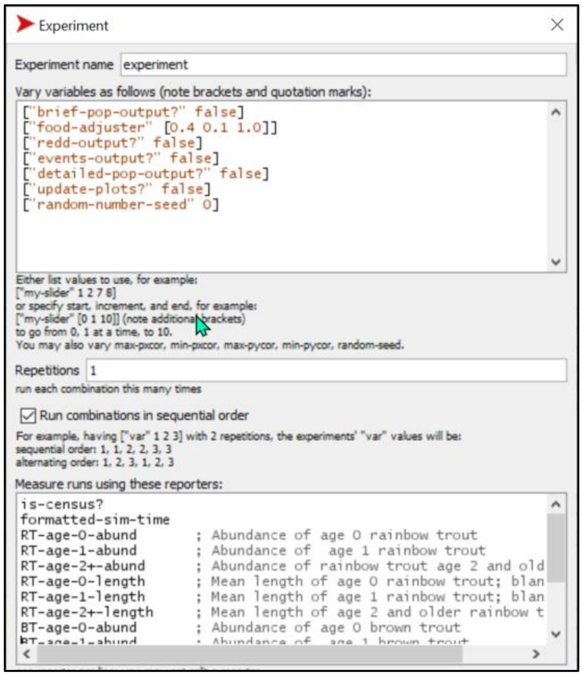
To get separate results for the two species, we need to change these to (this is one place where species names must be coded in):

```
  ;; BehaviorSpace outputs, updated in update-BehaviorSpace-outputs.
  RT-age-0-abund       ; Abundance of age 0 Rainbow Trout
  RT-age-1-abund       ; Abundance of age 1 Rainbow Trout
  RT-age-2+-abund      ; Abundance of Rainbow Trout age 2 and older
  RT-age-0-length      ; Mean length of age 0 Rainbow Trout; blank if none
  RT-age-1-length      ; Mean length of age 1 Rainbow Trout; blank if none
  RT-age-2+-length     ; Mean length of age 2 and older Rainbow Trout; blank if none
  BT-age-0-abund       ; Abundance of age 0 Brown Trout
  BT-age-1-abund       ; Abundance of age 1 Brown Trout
  BT-age-2+-abund      ; Abundance of Brown Trout age 2 and older
  BT-age-0-length      ; Mean length of age 0 Brown Trout; blank if none
  BT-age-1-length      ; Mean length of age 1 Brown Trout; blank if none
  BT-age-2+-length     ; Mean length of age 2 and older Brown Trout; blank if none
```

Then we need to modify the code that calculates the value of these variables, which is at the end of the Code tab in a special procedure called `update-behaviorspace-outputs`. We replace the code that calculates abundance and mean length by age class with the following code, which should be self-explanatory (the NetLogo command `let` creates a new temporary variable, while `set` changes the value of an existing variable—here, the global variables we created above). The case-sensitive species names used here must exactly match those in the parameter file.

```
    let RT-age-0s trout with [trout-age = 0 and trout-species = "Rainbow"]
    let RT-age-1s trout with [trout-age = 1 and trout-species = "Rainbow"]
    let RT-age-2s trout with [trout-age >= 2 and trout-species = "Rainbow"]
    let BT-age-0s trout with [trout-age = 0 and trout-species = "Brown"]
    let BT-age-1s trout with [trout-age = 1 and trout-species = "Brown"]
    let BT-age-2s trout with [trout-age >= 2 and trout-species = "Brown"]

    set RT-age-0-abund sum [trout-superind-rep] of RT-age-0s
    set RT-age-1-abund sum [trout-superind-rep] of RT-age-1s
    set RT-age-2+-abund sum [trout-superind-rep] of RT-age-2s
    set RT-age-0-length ifelse-value (any? RT-age-0s)
      [mean [trout-length] of RT-age-0s] [""]   ; Mean length; blank if none
    set RT-age-1-length ifelse-value (any? RT-age-1s)
      [mean [trout-length] of RT-age-1s] [""]
    set RT-age-2+-length ifelse-value (any? RT-age-2s)
      [mean [trout-length] of RT-age-2s] [""]

    set BT-age-0-abund sum [trout-superind-rep] of BT-age-0s
    set BT-age-1-abund sum [trout-superind-rep] of BT-age-1s
    set BT-age-2+-abund sum [trout-superind-rep] of BT-age-2s
    set BT-age-0-length ifelse-value (any? BT-age-0s)
      [mean [trout-length] of BT-age-0s] [""]   ; Mean length; blank if there are none
    set BT-age-1-length ifelse-value (any? BT-age-1s)
      [mean [trout-length] of BT-age-1s] [""]
    set BT-age-2+-length ifelse-value (any? BT-age-2s)
      [mean [trout-length] of BT-age-2s] [""]
```

After these changes, you can add all the output variables to the reporters used to measure your BehaviorSpace experiment (Figure 66).



Experiment name experiment

Vary variables as follows (note brackets and quotation marks):

```
["brief-pop-output?" false]
["food-adjuster" [0.4 0.1 1.0]]
["redd-output?" false]
["events-output?" false]
["detailed-pop-output?" false]
["update-plots?" false]
["random-number-seed" 0]
```

Either list values to use, for example:
["my-slider" 1 2 7 8]
or specify start, increment, and end, for example:
["my-slider" [0 1 10]] (note additional brackets)
to go from 0, 1 at a time, to 10.
You may also vary max-pxcor, min-pxcor, max-pycor, min-pycor, random-seed.

Repetitions 1
run each combination this many times

☑ Run combinations in sequential order

For example, having ["var" 1 2 3] with 2 repetitions, the experiments' "var" values will be:
sequential order: 1, 1, 2, 2, 3, 3
alternating order: 1, 2, 3, 1, 2, 3

Measure runs using these reporters:

```
is-census?
formatted-sim-time
RT-age-0-abund       ; Abundance of age 0 rainbow trout
RT-age-1-abund       ; Abundance of  age 1 rainbow trout
RT-age-2+-abund      ; Abundance of rainbow trout age 2 and old
RT-age-0-length      ; Mean length of age 0 rainbow trout; blan
RT-age-1-length      ; Mean length of age 1 rainbow trout; blan
RT-age-2+-length     ; Mean length of age 2 and older rainbow t
BT-age-0-abund       ; Abundance of age 0 brown trout
BT-age-1-abund       ; Abundance of  age 1 brown trout
```

Figure 66. BehaviorSpace experiment setup with additional output variables reporting separate results for Rainbow and Brown Trout.

17 Troubleshooting Guide

Here we maintain a list of problems often encountered with the inSTREAM 7 software, with potential solutions. Additional troubleshooting information may be available at the inSTREAM web site described in Sect. 1.1.

Symptom: When I try to open and set up inSTREAM, I get an error message saying that my model is too large to run in the memory available to NetLogo.

(This issue and the solution presented here are also addressed in the Frequently Asked Questions section of NetLogo's User Manual.) By default, NetLogo allocates one gigabyte of RAM to itself when it starts up. This amount can be too small when modeling large spaces or when the value of ***world-resolution*** is too small. See Sect. 14.1.5 for information on setting ***world-resolution*** to an appropriate value.

Many inSTREAM users will need to increase the memory allocation to NetLogo. Doing so can make larger BehaviorSpace experiments feasible; and (on at least some hardware) NetLogo can become very slow when it needs most of the available memory. To allocate more memory to NetLogo, go to the NetLogo program's app directory (in Windows, it is typically `C:\Program Files\NetLogo 6.2\app`) and find a file named `NetLogo.cfg`. Open this file in a text editor—you will likely need administrator privileges to change this file, so open the editor in administrator mode—and find these lines:

```
[JVMOptions]
-Xmx1024m
```

The second of these lines allocates one gigabyte (1024 megabytes) of RAM to NetLogo. Modern computers typically have much more RAM than this available; you can change this parameter to a substantial fraction (e.g., ¼) of your computer's total RAM. One limitation to remember is that if you typically run several copies of NetLogo at the same time (e.g., running several projects), each will use up the allocation set here. To set the RAM allocation to, e.g., 4 gigabytes, use `-Xmx4096m`. For a high-performance computer that lets you execute dozens of BehaviorSpace runs at once, you may need to set the allocation to 10 or 20 gigabytes.

(This RAM allocation applies to each instance of NetLogo. If you open several projects, each in its own NetLogo window, each will use the amount of RAM specified in the `NetLogo.cfg` file. Consequently, making this allocation large can limit how many copies of NetLogo can be open at once.)

Symptom: When I try to set up inSTREAM, I get an error statement indicating there were no wet cells at initialization.

This error statement is designed to make sure that, on the first simulated day, initial trout have wet cells to occupy. Occasionally, after editing the parameter file in NetLogo, this error statement mysteriously occurs without justification. Simply click the setup button again and the error should not recur (unless there is a real problem with initialization).

Symptom: When I set up inSTREAM I get an error statement that includes:

```
OF expected input to be a turtle agentset or turtle but got the number 0
instead.
```

This error can be caused by the value of the parameter ***world-resolution*** being too small; see Sect. 14.1.5 for guidance on setting its value.

Symptom: When I click on the setup button, it takes an extremely long time for the computer to finish reading GIS files and building cells.

You may also get error statements indicating that NetLogo ran out of memory while the executing `GIS:APPLY-COVERAGE`. This stage of setup can take a long time (while using all your computer's processors) when simulating a large reach (or reaches). However, this problem is more likely to result from using a value of ***world-resolution*** that is too small. See Sect. 14.1.5 about finding a good value of this parameter.

Symptom: When I set up inSTREAM I get an error statement that includes:

```
error (IllegalArgumentException)
 while observer running ASK
  called by procedure SET-UP-TIME
  called by procedure SETUP
  called by Button 'setup'

NetLogo is unable to supply you with more details about this error.  Please
report the problem at https://github.com/NetLogo/NetLogo/issues, or to
bugs@ccl.northwestern.edu, and paste the contents of this window into your
report.
```

```
java.lang.IllegalArgumentException: Invalid format: "1/1/1989" is too short
```

This error occurs when the first column of at least one row of a time series input file (Sect. 13.5) is not in the proper date-time format. This often happens when software used to maintain the file, such as Excel, changes the date and time to its default format when you open the file. You may have to re-open the file, set the date-time format manually, and re-save the file.

Symptom: When I set up inSTREAM I get an error statement that includes:

```
java.lang.IllegalArgumentException: Invalid format: "; Flow input..."
```

where the text in quotation marks is a comment in my time series input file.

This error is because a quotation mark precedes the semicolon that identifies comment lines at the top of this file. This file is in CSV format, and the standard for CSV files includes that any values can be inside quotation marks. Some software therefore puts quotation marks around text values when saving a file in CSV format. However, the code in inSTREAM 7 that reads time series input files does not fully comply with this standard and cannot accept quotation marks. (This problem may be corrected in future releases.)

Simply edit the input file in a text editor (not Excel) and delete all quotation marks.

Symptom: When I set up inSTREAM I get an error message that reads:

```
error (ArrayIndexOutOfBoundsException)
 while observer running ASK
  called by procedure SET-UP-TIME
  called by procedure SETUP
  called by Button 'setup'
...
java.lang.ArrayIndexOutOfBoundsException: 0
 at
time.datatypes.LogoTimeSeries.parseTimeSeriesFile(LogoTimeSeries.java:152)
 at time.datatypes.LogoTimeSeries.<init>(LogoTimeSeries.java:39)
 at
time.primitives.TimeSeriesPrimitives$TimeSeriesLoadWithFormat.report(TimeSeri
esPrimitives.java:150)
```

This error is probably because the time series input file contains blank lines or blank columns. When input files are maintained or edited in Excel and then saved in CSV format, Excel very often adds blank lines or columns that cause this problem.

Open the file in a text editor (not Excel) and remove extra lines, including any rows that contain only commas—most likely, at the end of the file. Extra columns appear as extra commas at the ends of data lines. Use search and replace to find and delete any pairs of commas with nothing between them (replace `",,"` with `""`).

Symptom: When I set up inSTREAM I get an error message that reads:

```
Extension exception: Extension could not parse input: ; (text from time
series input file header rows)(M/d/yyyy H:mm)
error while reach 1 running TIME:CREATE-WITH-FORMAT
```

or

```
Fish initialization file CC3A-InitialPopulations.csv has invalid fish age in
row: [; Trout initialization input for InSTREAM-7      ]
```

or some other error indicating that inSTREAM could not read an input file.

These error messages indicate that NetLogo could not read an input file properly. First, check for obvious mistakes such as missing semicolons at the start of comment lines. But these errors also commonly occur the file was saved in Excel's "CSV UTF-8" file type, which NetLogo cannot read. Instead, save the file in "CSV (Comma delimited)" file type.

(NetLogo actually requires text input files to be in UTF-8 encoding, but Excel's "CSV UTF-8" option apparently actually uses format called "UTF-8-BOM", which NetLogo cannot read. You can open a file in Notepad++, a widely used free editor for plain text, and use Notepad++'s "Encoding" menu to see a file's format and change it to "UTF-8".)

Symptom: InSTREAM executes setup but when I start a simulation I get an error message:

```
'temperature' is not a column in this time series
error while reach 0 running ERROR
```

(or `'flow'` or `'turbidity'`). However, my time series input file includes this variable.

This error message occurs when there is a space or other invisible character in one of the column labels. Open the time series input file in a text editor (not Excel) and delete any spaces, tabs, etc., within the column labels. The row of column labels must look exactly like this:

```
Date,flow,temperature,turbidity
```

(except that the order in which flow, temperature, and turbidity appear can vary).

Symptom: When I set up inSTREAM I get an error message that reads:

```
Depths file Upstream-Depths.csv has a row with cell ID not in GIS file:
error while reach 0 running ERROR
  called by procedure READ-HYDRAULICS
```

If the error statement does not provide an ID code for the problem cell, this error is probably because of blank lines at the end of a depth or velocity input file (Sect. 13.3). Open the file in a text editor (not Excel) and remove extra lines: go to the end of the file and delete any lines that contain only commas.

Symptom: When I try to set up inSTREAM, I get an Extension exception called "Unsupported projection".

This error means that you saved your GIS shapefile using a projection that is not supported by NetLogo's GIS extension. See Sect. 13.2 for information on projections and ways of overcoming this problem. If your shapefile uses a projection similar to one of the supported projections, you may be able to trick inSTREAM into accepting it by editing the projection file (the file in your shapefile directory with the extension `.prj`). In that file, find the projection name (e.g., `PROJECTION["Transverse_Mercator"]`) and change the name of the projection to one of those supported by the NetLogo GIS extension. Save the file and see if inSTREAM can now use your shapefile. If not, you will need to save the shapefile with one of the projections supported by NetLogo.

Symptom: I ran a BehaviorSpace experiment using replicates, but the "replicates" produced exactly the same results.

This is most likely because you used a value of ***random-seed*** other than zero, in which case your value of ***random-seed*** supersedes BehaviorSpace.

There are two ways to get replicates—multiple model runs with all parameters and inputs the same but different random number sequences—in BehaviorSpace experiments. The simplest way is to set ***random-seed*** on your model Interface to zero, and then select the number of replicates you want in the BehaviorSpace experiment setup. The only limitation of this method is that you can never exactly reproduce the experiment because you have no control over the random number generator.

The second method is to set the value of ***random-seed*** as a variable you vary in the BehaviorSpace experiment. Simply add a line to the "Vary variables as follows" box that contains `["random-seed" 3 4 5 6 7]`. (Replace `3 4 5 6 7` with a sequence of your lucky numbers, with as many numbers as you want replicates.) Then each scenario in the experiment will be repeated with different random numbers, and the experiment can be exactly reproduced because you know the random seed for each model run. (If also using the year shuffler, see Sect. 15.5 about how random number seeds affect it.)

Symptom: I changed some parameter values and then ran a BehaviorSpace experiment, but BehaviorSpace seemed to use the old parameter values.

NetLogo and especially BehaviorSpace can sometimes get confused about changes in the parameter file and use old values (as discussed in Sect. 14). To ensure that edits in the parameter file are saved and used in all future model runs, save both the parameter file and the NetLogo file, close them both, and re-open the NetLogo file before running BehaviorSpace.

Symptom: My BehaviorSpace output always reports zero abundance for one (or several) species, when I use output variables like those described in Sect. 16.6.

This error is most likely because a species name in the code that calculates the output (as illustrated in Sect. 16.6) does not exactly match the species name in the parameter file (parameter ***species-list***; Sect. 14.3). These names are case sensitive.

Symptom: InSTREAM halts with an error statement saying velocity was interpolated to a negative value at one cell, even though it was simulating a period of high flow.

InSTREAM's method for calculating velocity from flow (Sect. 9.3) can produce a negative velocity when extrapolating to a flow higher than any in its lookup table. This can happen when, in the lookup table imported from a hydraulic model (sects. 9.2, 13.3), there are cells with lower velocity at the highest flow than at the second-highest flow. In that situation (which is not uncommon), the extrapolation method produces velocities that get lower, and potentially negative, as flow increases above the highest lookup-table flow.

The best solution to this problem is to make sure the hydraulic input lookup tables cover the entire range of flows used in the inSTREAM simulations, by running the hydraulic model at higher flows. If that is not feasible, there is a work-around: open the velocity lookup table in spreadsheet software and a use spreadsheet formula to identify the cells that have velocity at the highest flow less than their velocity at the second-highest flow. For only those cells, replace the highest-flow velocity with the velocity at the second-highest flow. This edit will cause inSTREAM to use the same velocity for any flow above the second-highest in the lookup table.

Symptom: InSTREAM does not create any fish during setup, or it does not initialize all species, or put fish in all reaches.

These problems are most likely due to a mistake in the fish initialization input file (Sect. 13.4). If the name of the trout species or of the reach does not exactly match (including capitalization) the species or reach names in the parameter file, then no fish will be created and no error statement raised.

Symptom: No fish spawn during simulations.

If a simulation runs normally but no trout ever spawn (or spawning is rarer than expected), it is almost certainly because the spawning criteria defined in Sect. 9.27 are rarely or never met. Often, the cause is an inappropriate value for the maximum flow for spawning (Sect. 9.27.5) or temperatures that are rarely within the allowed range (Sect. 9.27.4). Comparing the parameters for these spawning criteria to the time series inputs during the spawning period should make it obvious which criteria prevent spawning.

Users can also diagnose this problem by turning on the optional output file for spawning readiness, as explained in Sect. 12.3. This output reports why each simulated female trout does not spawn on each day. Be aware that this output can be very large.

Part IV: Application Guide

18 Introduction and Overview of the InSTREAM 7 Application Guide

18.1 Application objectives

Part IV of the inSTREAM 7 documentation provides guidance on applying the model to study sites and questions. Applications of inSTREAM best start with one or several specific objectives—questions of management or research relevance that the model application is designed to address. A clearly stated objective is essential for making good decisions about exactly how to apply any model (e.g., Sect. 25 discusses study designs for different objectives). Sects. 1.2 and 3.1 identify general kinds of objectives that inSTREAM is designed for; here we reiterate common application objectives.

Instream flow and temperature assessment has always been a primary purpose of inSTREAM. This objective typically is addressed by simulating alternative flow and temperature management scenarios and their effects on trout populations. In addition to using inSTREAM to compare and evaluate pre-determined management scenarios, simulation experiments can also contribute to the design of effective and efficient management by providing better understanding of site-specific effects of flow and temperature.

Design and evaluation of habitat restoration projects has been another common application of inSTREAM and related models. For example, Railsback et al. (2013) contrasted simulated salmon spawning and rearing success at two sites, one with and one without extensive habitat restoration. Other, unpublished, applications have simulated alternative channel restoration concepts to assess their relative benefits, prior to final design and construction.

Its ability to assess and compare the relative value of different potential management actions has proven a particularly useful characteristic of inSTREAM. Railsback et al. (2013) included, as part of their evaluation of a restoration project, simulation analyses evaluating the relative potential of improving salmonid populations by manipulating flow, summer or winter temperatures, feeding and hiding cover, and spawning gravel availability.

InSTREAM has also proven useful for a variety of other management and research objectives, including assessment of alternative angler harvest regulations (Ayllón et al. 2016, 2018, 2019a, 2021), prediction of climate change effects (Ayllón et al. 2016, 2019b), understanding watershed-level effects of movement barriers (Harvey and Railsback 2012, 2021), exploring the population-level effects of turbidity (Harvey and Railsback 2009), assessing population and community effects of sublethal contaminants (Forbes et al. 2019), and understanding the concept of “food limitation” (Railsback and Harvey 2011).

18.2 Typical application steps and summary of data requirements

The rest of Part IV is dedicated to guidance on the individual steps of applying inSTREAM. This section provides an overview of these steps and the input needed for each.

Regardless the specific study objectives, inSTREAM 7 applications typically involve:

1. Deciding how many reaches to model and where to place them.
2. Developing time series of the variables that drive simulations: flow, temperature, and turbidity. These time series typically are daily values for sufficient years to represent the scenarios being evaluated. Important differences between inSTREAM and simpler flow evaluation methods include that inSTREAM: considers the entire flow regime, not just minimum flows; considers year-round conditions instead of only the times (e.g., summer) presumed most stressful; and considers the interacting effects of flow, temperature, and turbidity. Consequently, it requires these time-series inputs for each reach and scenario. The inputs are typically developed from a combination of existing gages, site-specific data collection, and modeling.
3. Assembly or collection of site-specific trout population data. InSTREAM is not heavily dependent on site-specific population data and can be used without it, but such data are desirable for defining initial conditions and calibrating the model.
4. Hydraulic modeling, including collection of reach bathymetry and calibration data. InSTREAM can be used with any two-dimensional or pseudo-two-dimensional hydraulic model that simulates how depth and velocity vary over space with flow.
5. Cell delineation and estimation of static habitat variables. The boundaries of the habitat cells that represent within-reach variation in habitat must be defined, either in the field or via GIS. Then users must develop values of each cell's variables representing velocity shelter for drift feeding, distance to escape cover, number of hiding places, and spawning gravel availability. Evaluating these variables typically requires a combination of field observations and GIS analysis.
6. Evaluation of site- and species-specific parameter values. InSTREAM has many parameters, but relatively few have site- or species-dependent values that need to be reconsidered in each application.
7. Model calibration. Calibration usually requires a few simple simulation experiments that vary a small number of particularly important yet uncertain parameter values and compare results to observed trout abundances and sizes.
8. Designing, executing, and analyzing the simulation experiments that address the study objectives. This is the most important step and requires considerably more time and effort than analysis of simple models such as PHABSIM. InSTREAM produces many kinds of output; understanding how and why its outputs respond takes work, but typically teaches us a great deal.

19 Study Reach Selection and Layout

Selecting the number, location, and size of study reaches is typically the first critical step in applying inSTREAM. The following considerations provide some guidance. Here, "site" refers to the longer stream length (or network) that the model represents. One or more reaches are often used to represent a site: for example, three reaches might represent a site that extends from a reservoir downstream to the confluence with a larger stream. Reach selection is typically viewed as a tradeoff between the benefits of representing more habitat and its costs (in money and time) for data collection, hydraulic modeling, and executing and analyzing simulations.

19.1 Reach selection

Selecting the number, size, and location of reaches requires judgment and consideration of the questions that inSTREAM is being used to address. We addressed this issue in an analysis of inSALMO (a version of inSTREAM for salmon) applied to 12 sites on one stream (Railsback et al. 2015). That analysis found that some model processes such as effects of temperature regimes are relatively insensitive to spatial extent because they do not vary much over space, while other mechanisms such as growth and survival are highly dependent on local habitat conditions and hence sensitive to how thoroughly space is represented. Relatively long reaches that typify overall habitat, combined as needed with small reaches that represent unique habitat "hot spots", were found effective at representing an entire system. Predictably, the influence of stochasticity on population dynamics varies inversely with reach size.

If the purpose of an inSTREAM application is to assess effects of river management—especially, flow, temperature, or turbidity regimes—then representativeness is a primary consideration in choosing the spatial extent: does the simulated habitat capture the range and approximate distribution of habitat types of the real stream?

If instead the purpose is to evaluate or design habitat at one site, such as a channel restoration site, then the model's spatial extent can be restricted to that site (e.g., Railsback et al. 2013). In such cases, though, the model's results will not reflect the buffering effects of movement into and out of the site and therefore can provide an index of site habitat quality, but not useful predictions of abundance.

InSTREAM can also be used to assess watershed-scale effects of factors such as turbidity and passage barriers (e.g., Harvey and Railsback 2007, 2012). In such applications, the primary spatial extent concerns are to capture the range and distribution of habitat across the watershed, in its various tributary sizes, and to represent enough space to prevent exaggeration of local extinction frequency.

For studies intended to assess the effects of river management, including instream flow studies, the primary consideration in selection of study reaches should be that reaches adequately represent the habitat diversity of the entire site. The number and location of reaches can be chosen to include any major differences in channel morphology within the site. Ideally, reaches will include all the major habitat types (pools, riffles, cascades, etc.) and also all cover features (wood, undercut banks, aquatic and overhead vegetation) in roughly the proportions they occur in the full site.

Habitat need not be excluded from the study reach because it has complex hydraulics that make hydraulic modeling difficult or uncertain. Errors due to modeling complex hydraulics are likely to be less important than those that would result from ignoring important habitat.

Spawning habitat deserves consideration in reach selection. Because inSTREAM does not incorporate immigration (except for movement among reaches), it can exaggerate the variability of annual reproduction when a site is represented by reaches containing little spawning habitat. The number of adults that spawn is modeled stochastically, and one often-important cause of redd mortality (scour) is also highly stochastic. This stochasticity tends to be higher for smaller reaches with fewer trout or little spawning habitat. Exaggerated variability in reproduction may particularly affect studies that address population persistence.

19.2 Reach layout in inSTREAM

Sites represented by multiple reaches can be represented as one inSTREAM project or as multiple projects. Isolated reaches that fish cannot move among can be represented as separate projects with their own inputs and calibration. Reaches that model trout can move among are best represented within a single project. Doing so requires assembling a GIS shapefile that includes all the reaches in a project (Sect. 13.2).

Laying out a site's reaches in GIS requires translating and perhaps rotating them so that their ends coincide (as in Figure 44). InSTREAM's methods for determining how far model trout can move in habitat selection use distances in the GIS space, so multiple reaches must be placed artificially adjacent to each other in that space if simulated trout are to be able to move among reaches. The distance between reaches in the inSTREAM input can be manipulated to affect what size trout can move among reaches. If the upstream end of one reach is dragged in GIS to be immediately adjacent to the downstream end of another reach, then all but the smallest fish will be able to move among the reaches. If the reaches are placed farther apart, then only larger fish will be able to move between them because the distance a trout can move increases with its length (Sect. 9.13.1).

20 Time Series Inputs

Time series of daily values of flow, temperature, and turbidity are the main environmental "drivers" of inSTREAM. If such time series are not already available, they must be developed, which usually requires data collection. Therefore, attention should be given to this input from the start of a model application.

Unlike previous versions, inSTREAM 7 does not require a specific (daily) temporal resolution in the time series inputs; input can be hourly, daily, weekly, or at irregular intervals. The model simply uses the input from the time closest to the middle of each time step (Sect. 8.1). However, all three variables are input at the same resolution.

InSTREAM simply reads in values for the time series inputs; these values must be prepared by the user for each reach and for each scenario when scenarios differ in the time-series inputs (e.g., when contrasting alternative instream flow regimes). Time series inputs are usually developed through a combination of collecting field data and other methods such as estimating flows with a reservoir or hydrologic model or by adjusting them from a nearby gage; modeling water temperature (for scenarios different from observed conditions) using physics-based simulation or statistical modeling; and statistical modeling of turbidity as a function of flow. Such methods are widely used; here, we provide guidance specific to inSTREAM.

If the model application includes comparison of alternative flow and temperature regimes, presumably including new regimes that have not occurred in reality, then some kind of modeling is necessary to develop the input representing each alternative regime. Predicting flows and temperatures under different management alternatives is a standard part of instream flow assessment, but often the focus is only on specific times of year, such as when flows are lowest and temperatures highest. Because inSTREAM evaluates year-round regimes, the models and data used to produce inputs must adequately represent all seasons and the full range of conditions that can be expected.

Turbidity is generally not monitored as easily or commonly as temperature or flow, and it is often tempting to ignore turbidity. (Input for projects with negligible turbidity, or with turbidity intentionally neglected, is discussed in Sect. 8.1.) However, the strong and well-defined effects of turbidity on trout feeding and predation risk are included in inSTREAM; turbidity can strongly affect predicted trout population responses to flow, temperature, and other habitat characteristics. These effects are most sensitive to turbidity values in an intermediate range of approximately 5 to 50 NTU; below that range turbidity has little effect, while effects become asymptotic at high values. Except for sites with consistently low turbidity (e.g., downstream of a large reservoir), it is important to develop reasonably accurate input for this intermediate range.

Where turbidity is caused primarily by suspended sediment, turbidity can be modeled as an increasing function of flow; we often assume turbidity is proportional to flow (e.g., Harvey and Railsback 2009). In some systems, however, phytoplankton can elevate turbidity at low flows. Turbidity can also vary seasonally, as erosion rates, algal production, and sediment settling rates vary with factors such as temperature and snow cover. Therefore, collecting grab samples over a wide range of flows and seasons is better than having no site-specific turbidity information at all. Grab samples can be used to fit rough but useful models of how turbidity varies with flow, season, and temperature.

21 Hydraulic Modeling and Input Preparation

21.1 Hydraulic modeling considerations

The hydraulic input used by inSTREAM 7 (Sect. 13.3) can be developed in many ways. We expect that it will most commonly be produced via two-dimensional hydrodynamic modeling, as that approach is now widely used in river management. However, the input could also be developed using traditional transect-based, pseudo-two-dimensional models such as the original PHABSIM hydraulic models, or even from direct field measurements. Because river hydraulic modeling expertise is now relatively widespread, here we focus on issues specific to inSTREAM and two-dimensional hydraulic modeling.

Because inSTREAM models the full flow regime, the hydraulic input must represent not just minimum or low flows but also the moderate and high flows represented in the input scenarios. For most inSTREAM applications, the emphasis remains on lower flows, so hydraulic model calibration and validation is also focused on lower flows. Often, also, cell-scale hydraulics are more challenging to model at lower flows, and inSTREAM is likely less sensitive to hydraulics at very high flows. However, keep in mind that when inSTREAM simulates flows above (or below) those included in the input imported from the hydraulic model, it simply extrapolates cell depths and velocities. No matter how uncertain the high-flow calibration of the hydraulic model, it is likely more accurate than inSTREAM's extrapolation; if we must extrapolate to flows above (or below) those we have calibration data for, it is likely better to do so with the hydraulic model instead of in inSTREAM. Collection of calibration data at moderate and high flows can improve inSTREAM applications by providing the ability to model such flows with reasonable accuracy.

A second hydraulic modeling consideration is that inSTREAM explicitly uses spatial resolutions (cell sizes) that are more aggregated than many river habitat hydraulic model calibrations. Hydraulic modelers sometimes assume that the finer their hydraulic modeling mesh, the better the model will represent fish habitat; however, that assumption is only true if the biological component of the model is explicitly designed for the same resolution. InSTREAM is designed for cell depths and velocities to represent averages over relatively large cells, while fine-resolution habitat elements such as velocity shelter for drift feeding are represented as cell parameters that the hydraulic model need not capture.

One way to match the hydraulic model's spatial resolution to that of inSTREAM is to use a relatively coarse hydraulic modeling mesh, so that mesh elements can correspond with inSTREAM cells. Another approach is to model hydraulics at a scale appropriate to represent hydraulics well, often finer than the inSTREAM cells, and then use GIS to aggregate hydraulic model results into cell-averaged values for inSTREAM. We often model reach hydraulics independently, using a relatively fine resolution, before delineating the inSTREAM cells. The hydraulic model results can in fact be part of the information used to delineate cells (Sect. 22). It is not advisable to use a hydraulic modeling mesh coarser than the inSTREAM cells.

Finally, in hydraulic modeling keep in mind the potential need to model how bed shear stress varies with flow, as part of the redd scour submodel (Sect. 23.1). Some hydrodynamic models can be used to estimate the relation between reach-average shear stress and flow needed for that submodel.

21.2 Preparation of hydraulic input

As input, inSTREAM requires tables of cell depths and velocities at a wide range of flows (Sect. 13.3). Preparing this input requires selecting the flows and simulating them in the hydraulic model. We typically include 20-40 flows in the input tables, spanning the full range of flows that could be in the scenarios to be modeled (Sect. 9.3).

Which flows to simulate in the hydraulic model and include in the inSTREAM input is an important decision. In general we include more low flows and fewer high flows in the input, because (a) most inSTREAM applications focus on evaluating scenarios that differ in low flows (e.g., evaluating minimum flow releases), (b) low flows typically are more common than high flows, and (c) as flow increases, depths and velocities usually become more uniform, so interpolation between flows becomes less uncertain. A general approach to selecting flows to simulate in the hydraulic model is to evenly space them by exceedance values: analyze a historic flow record and simulate (e.g.) flows exceeded on 0.1%, 5%, 10%, 15%, 20% ... 95%, and 99% of days.

The final step in hydraulic input preparation is calculating characteristic depths and velocities for each habitat cell, for each flow simulated in the hydraulic model. When the hydraulic model mesh and the inSTREAM cells are the same, cell values are simply equal to the depth and velocity of the corresponding hydraulic model node. However, when the hydraulic model uses a finer spatial resolution than inSTREAM does, we must re-scale hydraulic model results to the inSTREAM cells. We do this via GIS: overlaying the inSTREAM cells and hydraulic model results and calculating a spatially averaged depth and velocity for each cell, at each of the simulated flows. GIS offers approaches such as weighing the values from hydraulic model nodes by how much of the cell's area each node represents.

22 Cell Delineation and Habitat Variables

Cell delineation is the task of defining the boundaries of inSTREAM's habitat cells. A closely related task is establishing values of the cell habitat variables that vary among cells but not over time (Sect. 4.1.3).

22.1 Delineating cell boundaries

The goals in cell delineation are the same regardless the method used. The primary goal is to delineate cells that capture important spatial variation in hydraulic conditions and other habitat variables as differences among cells, while minimizing variation within each cell. In other words, cells should be placed so that each bounds an area of relatively homogenous habitat while sharp gradients in habitat occur at cell boundaries. This goal cannot always be achieved where habitat gradients are steep, such as along a steeply sloped bank.

A second important goal is for cell delineation to produce a useful spatial resolution, as discussed in Sect. 4.2.1. Cells should be no smaller than necessary to capture important habitat gradients, and never smaller than the area over which a large trout or a superindividual representing multiple small trout might typically feed: we try to avoid cells less than 2-4 m^2 in area. Cells smaller than this can cause artifacts in inSTREAM results, and large numbers of unnecessarily small cells substantially increase model execution time.

In traditional instream flow modeling, cells are delineated in the field by placing transects across the stream, selecting the cell widths, and assuming the distances upstream and downstream represented by the transect. This approach can be used with inSTREAM, especially in combination with traditional PHABSIM (Bovee et al. 1998) hydraulic models (Figure 67, top panel). Railsback et al. (2009) provide important guidance on this approach, including ways it must be used differently for inSTREAM than for PHABSIM.

InSTREAM, unlike PHABSIM, depends on an explicit map of stream habitat and its spatial relations. Modern two-dimensional hydraulic models allow us to produce less distorted maps than transect-based approaches. Therefore, most inSTREAM applications now use cells that are irregular polygons that follow the actual stream geometry and habitat closely (Figure 67, middle panels).

Polygonal cells that follow actual stream topography and habitat can be difficult and time-consuming to delineate in the field. With modern GPS equipment the methods described below for delineating cells in GIS could also be used in the field, but doing so would require a great deal of field time for large sites. Field delineation also makes it infeasible or especially time-consuming to view results and then correct problems while still in the field.

The approach we most often use now, especially for large streams, is to delineate cell boundaries in GIS while viewing and considering a variety of spatial information. This approach starts with assembling GIS coverages of the reach with:

- Bed topography, such as bed elevation contours;
- Depth and velocity distributions, as contours or shading of the hydraulic model mesh at one or several characteristic flows;
- High resolution aerial or satellite photography; and

- Field observations of cell habitat variables (discussed below).

An analyst familiar with the site and inSTREAM can then create the habitat cells in the GIS while looking at these coverages. We have used two methods. One, illustrated by the second panel of Figure 67, is for the analyst to click in the corners (vertices) of each cell polygon. This method allows complete control over cell boundaries, but it can be slow and painstaking, and is subject to the accidental creation of small sliver cells and small gaps between cells. The second method is for the analyst to click in cell centroids and then use the GIS to create cells as Thiessen (Voronoi) polygons around those centroids. The analyst must try to place the centroids so that the Thiessen polygon boundaries fall near breaks in hydraulic and habitat conditions, and the method typically involves trial and error—it is easy to delete a polygon and reposition its centroid, then re-create the polygons. The third panel of Figure 67 shows a reach with cells created this way.

Dudley (2018) developed a fourth cell delineation method, specifically for adapting large-scale and coarse hydraulic simulations of large rivers to models like inSTREAM. That method imposes a uniform cell grid to represent a hydraulic model mesh that typically is coarser (Figure 67, bottom panel). This method can be useful in the absence of detailed bathymetric and hydraulic data, but provides less resolution for representing small areas of important habitat, e.g., shallow channel margins for small juveniles. Dr. Dudley (Southwest Fisheries Science Center, National Marine Fisheries Service, Santa Cruz, CA) makes software for generating cells available as a plugin to the QGIS geographic information software, at https://github.com/pndphd/hydro_generator.

While the number of cells strongly affects model execution time, their shape or complexity has no such effect.

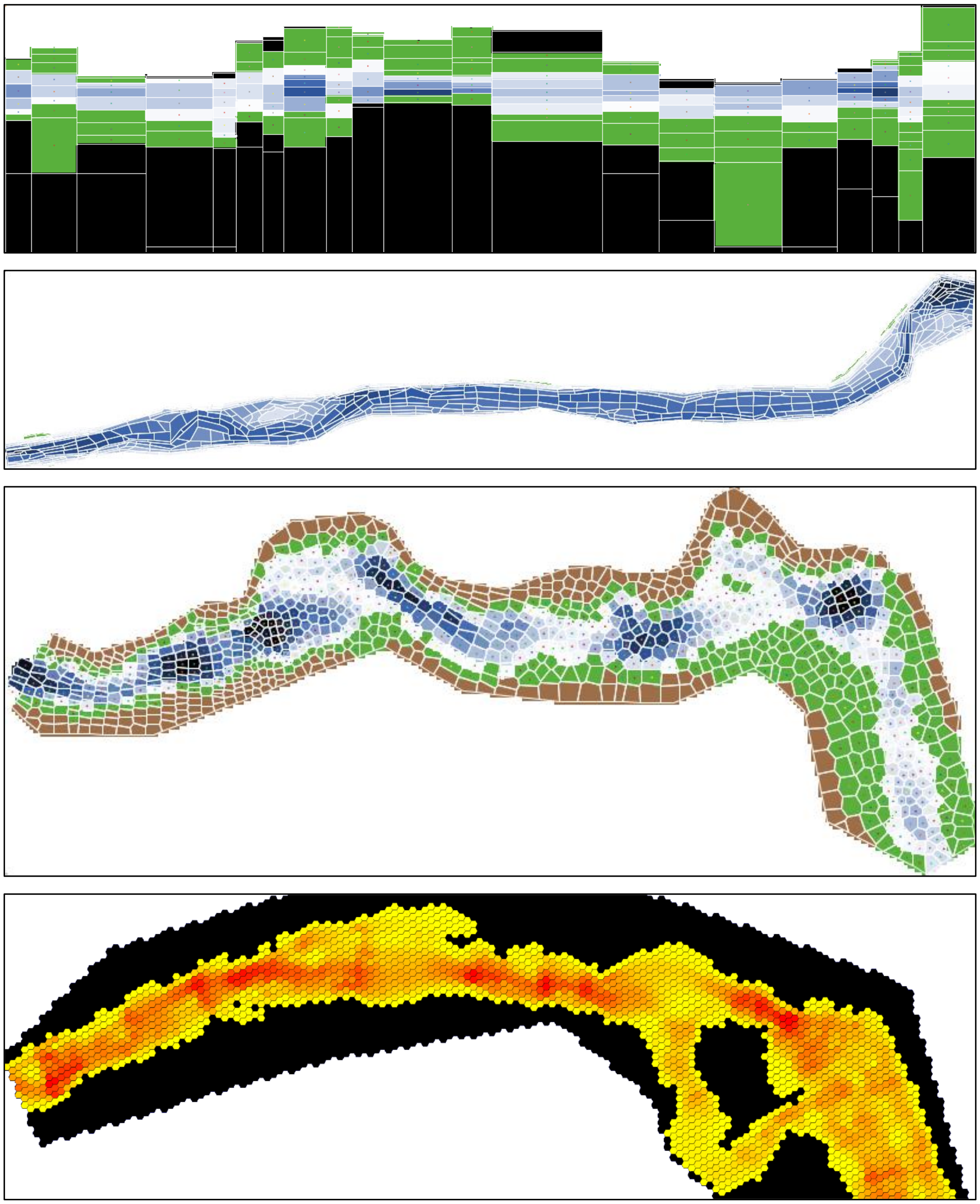


Figure 67. InSTREAM reaches developed with four alternative cell delineation methods. Top: traditional transect method with cells delineated in the field. Second: GIS delineation with cell vertices entered manually. Third: GIS delineation with cell centroids entered manually and boundaries defined as Thiessen polygons. The top reach is 184 m long, with 167 cells having mean area of 19 m^2; the middle reach extends 1355 m from east to west with 825 cells averaging 8.6 m^2; and the bottom reach extends 598 m east-west and has 832 cells with mean area of 48 m^2. Bottom: Hexagonal cells produced with the QGIS tools of Dudley (2018). Each cell has an area of 20 m^2, here shaded by velocity.

22.2 Assigning cell habitat variable values

Four of the habitat variables input for each cell (Table 24; see also Table 3) are based on field observations. Sect. 9 describes the effects of these variables; here we provide guidance on developing their values. First, we provide general guidance on field estimation of habitat variables, then we discuss what to observe for each variable, and finally we provide several methods for estimating values via field observation and GIS analysis.

We recommend linking the tasks of assigning cell habitat variable values and delineating cell boundaries: we must assign habitat variable values to each cell, but cell delineation should be informed by observations of the habitat variables.

22.2.1 General guidance on field estimation of habitat variables

The following advice for estimating habitat variable values in the field is largely based on recommendations by Railsback and Kadvany (2004, 2008) for judgement-based evaluation of stream habitat. None of the cell habitat variables can be defined very precisely, and variation in their values with flow and fish characteristics is ignored by inSTREAM. Therefore, the best way to evaluate them is simply by visual estimation by observers experienced with trout feeding, hiding, and spawning behavior. However, the following practices can make visual estimation more credible and reproducible:

- Use more than one observer, each with sufficient experience to have a good mental model of the kinds of habitat trout use for drift feeding, hiding, and spawning.
- In advance, thoroughly understand from the model description of Part II and the following subsection exactly what each variable means and how it is used.
- Discuss and define in advance how to interpret cell habitat variables for the study site, especially what size fish are represented and what kind of cover they are assumed to use for hiding and feeding. Develop a mental model of what each variable represents, in advance.
- Discuss and agree on observed values in the field, instead of having each observer record separate values and averaging them later.
- Focus on estimating cell-average values, recognizing that there is no single "right" value.

Estimation of habitat variables may best be conducted during low flows when more of the stream channel can be waded and observed. However, at extremely low flows many of the cells can be dry, which introduces additional uncertainty into the estimates. Estimation of these values can be especially challenging in large and turbid streams.

22.2.2 Cell habitat variables

Here we remind users what the field-estimated cell habitat variables represent and provide information for observing them.

Cell-escape-dist is a characteristic distance (m) from feeding sites to escape cover (Sect. 9.18.5). Trout use escape cover to elude predators by becoming less visible or harder to catch. Escape cover can be provided by vegetation, rocks, woody debris, undercut banks, and shade. The value of *cell-escape-dist* for a cell is an estimate of how far a trout would have to move from a typical feeding location to the nearest useful escape cover. The nearest cover need not be in the same cell.

Where escape cover is sparse, this variable can be estimated by identifying likely feeding locations and the nearest escape cover, and estimating the distance between them. In places such as heavily vegetated floodplains, macrophyte beds, or where crevices are abundant, we often assume that escape cover is essentially everywhere and set *cell-escape-dist* to zero or <1 m.

Cell-num-hiding-places is the number of places where adult trout can conceal themselves when not feeding (Sect. 9.18.6). InSTREAM treats concealment cover as discrete places where trout can be visually isolated from each other. The first step in evaluating *cell-num-hiding-places* is to identify a range of trout sizes that values are assumed applicable to. Generally, we recommend using a typical size for a small adult trout as the minimum that concealment cover must accommodate.

Concealment cover is typically provided by interstitial space among substrate items and undercut banks. Many kinds of cover that can provide temporary escape do not provide concealment over longer periods. Estimating the number of hiding places of course requires guesswork because we cannot see into hiding places to determine how many fish they could conceal. We typically estimate one hiding place for each isolated location (e.g., a boulder that appears to have space at its base), and several hiding places for a larger area complex enough to isolate several individuals (e.g., an undercut bank with roots).

Cell-frac-vel-shelter is the fraction (0.0 to 1.0) of the cell providing velocity shelter for drift feeding (Sect. 9.13). Velocity shelter is typically provided by large substrate items and sometimes by velocity shear zones that allow a trout to swim in lower velocities than it captures food from. Defining useful velocity shelter depends on the size of trout being considered: small juveniles may gain useful reductions in swim speed from small substrate or bed roughness that would not benefit adults. Except for inSTREAM application specifically focused on juvenile rearing, we quantify velocity shelter as area that would benefit small adults and larger trout.

Cell-frac-spawn is the fraction of the cell with substrate suitable for spawning (Sect. 9.29). Delineating spawning substrate requires considering both substrate size and the size of substrate patches. The range of gravel sizes useful for spawning can depend on spawner sizes and (perhaps) substrate availability (Kondolf 2000): ideal spawning gravels are relatively free of fines or loose enough that the spawner can clean fines away, large enough to promote interstitial flow, and small enough for the female to move. When such substrate is available in relatively large beds, *cell-frac-spawn* can be evaluated as the fraction of a cell within such a gravel bed. However, in stream reaches lacking beds of ideal gravel trout may instead spawn in small pockets of gravel such as those accumulated behind boulders, which should therefore be captured in *cell-frac-spawn*.

22.2.3 Methods for observing and assigning values

Here we describe three methods for collecting field data on habitat variables and using the data to assign values to cells. Different methods may be used for different variables.

Estimating values for each cell in the field. This method relies on field observation instead of GIS analysis. Each cell’s habitat variables are evaluated in the field, which requires identifying cell boundaries. In fact, this method can combine cell delineation with habitat variable evaluation: the field observer team can set cell boundaries and evaluate habitat variables at the same time. This approach is most practical for small reaches and relatively regular cells; we used it successfully with transect-based rectangular cells (as in the top panel of Figure 67). However, with large reaches of nonuniform cells (the middle panels of Figure 67) the method becomes impractical due to the number of cells and the difficulty of identifying their boundaries in the field.

Mapping habitat elements. This method involves mapping specific habitat elements—escape and concealment cover, zones of velocity shelter, and patches of spawning gravel—in the field and then assigning variable values to cells in GIS. Appropriate GIS methods can vary among habitat variables. For example, when the locations of escape cover are mapped, GIS analysis can estimate *cell-escape-dist* for each cell by calculating the distance to nearest escape cover for each point in a grid laid over the cell. The values of *cell-frac-vel-shelter* and *cell-frac-spawn* can be calculated as the fraction of cell area that overlaps mapped areas of velocity shelter and spawning gravel.

Mapping areas of similar habitat. Using this method, field observers map patches of habitat that are relatively uniform for the inSTREAM habitat variables, and then estimate variable values for each patch. These patches could include, e.g., patches of gravel bar, cobble bed, riffle, and boulder garden. Patches can be delineated by collecting GPS locations around their perimeter or simply by drawing them on detailed maps or air photos. Then, in GIS, these patches are digitized as “habitat variable polygons” and laid over the habitat cells. Cell boundaries can be selected or adjusted to coincide with the boundaries of habitat variable polygons, so that cells better capture habitat variation. Cell habitat variable values are assigned as the values of the habitat variable polygon that contains the cell centroid. (Habitat variables are typically discontinuous, so spatially averaging values may not produce a representative value.)

This approach of mapping areas of similar habitat has been the most useful for large study sites where the other approaches would require excessive field effort. It works best at sites large enough to have patches of relatively similar habitat that are substantially bigger than the cells and relatively easy to delineate in the field. Another benefit of this approach is that the observed habitat variable polygons are useful aids when delineating cells.

23 Parameter Evaluation

This section addresses which of inSTREAM 7’s habitat and trout parameters should have their values reconsidered and perhaps altered for application of the model to a new site. Only a few parameters typically need adjustment. Part II provided guidance on selecting values for parameters, with the description of each parameter’s meaning and use. Look here to learn which reach and trout parameters should be reconsidered, then find those parameters in Part II (especially, Sect. 9) to see how their values are determined.

Many inSTREAM parameters define functions. Before changing the value of such parameters, we strongly recommend users implement the function by itself and fully understand how the change affects the function, because modification of parameters can change nonlinear functions in unexpected ways. For logistic function parameters, it is easy to implement the function in a spreadsheet from the equations in Sect. 2.2.11, plot its output, and view how the function changes with parameter values.

In fact, a spreadsheet implementation of a logistic function can be very useful for finding reasonable parameter values. Users can assume two or more values of the function that seem reasonable (e.g., survival of terrestrial predators increased by 50% at a depth of 100 cm and 80% at 200 cm), then use a spreadsheet's "solver" tool to find parameter values that minimize the difference between the logistic function's value (here, relative survival at 100 and 200 cm depth) and the assumed values.

23.1 Reach-specific habitat parameters

Reach parameters represent reach-specific conditions, so most of them need to be re-evaluated when inSTREAM is applied to a new site or reach. Table 31 presents all reach parameters; grey shading indicates parameters especially likely to vary among sites. Be aware that the parameters typically used to calibrate inSTREAM (Sect. 24) are reach parameters.

In addition to the reach parameters, three global parameters are also site-specific. The parameter ***latitude*** is simply the site's latitude (degrees North); all reaches in a simulation are assumed to have the same latitude. ***Light-correction*** and ***light-at-night*** also apply to all reaches and could vary among sites with very different light conditions. None of these parameters appear to have strong effects on model results (Sect. 26.2).

Two parameters, ***reach-shear-A*** and ***reach-shear-B***, serve to relate reach-average Shields shear stress to flow in the redd scour submodel (Sect. 9.35). Before addressing these parameters, users must first determine if the redd scour submodel is appropriate for their study reach. The submodel is designed for alluvial streams with fairly extensive spawning gravel beds or bars. In reaches where spawning occurs primarily in pocket gravels behind obstructions, one might assume that scour processes can still be approximated with the submodel because in such reaches scour of redds is probably still a stochastic event that increases with streamflow and decreases with redd depth. Setting ***reach-shear-A*** to zero turns off scouring mortality.

The two shear parameters are typically evaluated by estimating a reach-scale Shields stress at each of several high flows and then fitting the parameters via logarithmic regression. Shields stress can be calculated from shear stress, which can be output from some hydraulic models. (Haschenburger 1999 provides the conversion in her Equation 3, but the method is widely used in the sediment transport literature.) This reach-scale Shields stress is intended to represent the shear stress in spawning areas. When calculating reach-scale Shields stress from hydraulic model output, we recommend averaging stress only over the within-channel cells where spawning is likely, not over the entire wetted channel, especially at high flows that inundate the flood plain. Averaging Shields stress over the entire wetted area under such conditions would underestimate stress in cells with redds.

When hydrodynamic models do not provide estimates of shear stress, Shields stress can be estimated using a standard equation relating it to hydraulic radius, energy slope, and sediment size, explained in Sect. 16.7.2 of Railsback et al. (2009). Because this method assumes that all flow resistance results from bed shear stress, Shields stress should be estimated for channel sections lacking sharp bends and obstacles that provide "form resistance" to flow.

The parameter ***reach-max-spawn-flow*** can be especially important to evaluate carefully because it can strongly limit spawning. This parameter represents the flow above which model trout do not spawn, which is essentially impossible to evaluate empirically. Instead, consider potential scour and dewatering redd mortality, and the flow regime. It is reasonable to assume that trout would not spawn at flows high enough to make redd scour likely, so ***reach-max-spawn-flow*** should be a flow with low probability of redd scour (Sect. 9.35), e.g., daily scour probability less than 0.05. This parameter should also prevent spawning at flows so high that suitable spawning locations (Sect. 9.29) are likely to be dry at flows typical during the redd incubation period. Finally, be sure that the value of ***reach-max-spawn-flow*** does not prevent spawning for much of the spawning period.

Table 31. Reach parameters.

Parameter	Value type	Typical value	Parameter meaning
reach-name	Text	`"Example-reach"`	The names of the reaches (or single reach). These names must exactly match the reach names used in the GIS shapefile.
time-series-input-file	Text	`"Example-reach-TimeSeriesInputs.csv"`	The name (and path, if necessary) for each reach's time series input file. More than one reach can use the same file.
depth-file-name	Text	`"Example-reach-depths.csv"`	Name (and path) for the files containing cell depth lookup input.
velocity-file-name	Text	`"Example-reach-vels.csv"`	Cell velocity lookup input file names.
reach-drift-conc	Number (often in scientific notation)	`3.75E-10`	Concentration of drift food (g/cm^3) for each reach (sects. 9.11.1, 9.22).
reach-search-prod	Number (often in scientific notation)	`1.0E-5`	Reach search food production rates (g/cm^2/d; sects. 9.11.1, 9.23).
reach-shelter-speed-frac	Real number between 0.0 and 1.0	`0.3`	Swimming speed of a trout using velocity shelter, as a fraction of the cell's velocity (Sect. 9.13).
reach-prey-energy-density	Real number	`2500`	Energy density of prey (trout food; J/g; Sect. 9.21).

Parameter	Value type	Typical value	Parameter meaning
reach-drift-regen-distance	Real number	`1000`	Distance (cm) over which consumed drift food is regenerated (Sect. 9.11.1).
reach-shading	Real number between 0.0 and 1.0	`0.9`	Fraction of sunlight that is *not* blocked by shading (by trees, hills, etc.; Sect. 9.10).
reach-fish-pred-min	Real number between 0.0 and 1.0	`0.9`	Daily probability of surviving predation by fish, for the most vulnerable trout (Sect. 9.19).
reach-terr-pred-min	Real number between 0.0 and 1.0	`0.95`	Daily probability of surviving predation by terrestrial animals, for the most vulnerable trout (Sect. 9.18).
reach-light-turbid-coef	Real number	`0.0017`	Multiplier coefficient in the equation for light attenuation as a function of depth and turbidity (Sect. 9.10).
reach-light-turbid-const	Real number	`0.0`	Constant coefficient in the light attenuation equation (Sect. 9.10).
reach-max-spawn-flow	Real number	`20`	Flow (m^3/s) above which trout do not spawn (Sect. 9.27.5).
reach-shear-A	Real number	`0.013`	Multiplier (s/m^3) in the equation calculating shear stress from flow, to model redd scour. A value of 0.0 turns off redd scour (Sect. 9.35).
reach-shear-B	Real number	`0.40`	Exponent (dimensionless) in the shear stress equation (Sect. 9.35).

23.2 Site- and species-specific trout parameters

Only a few of the trout parameters are usually re-evaluated among application sites, even when different trout species are modeled. In fact we discourage users from replacing our standard trout parameter values—except those specifically listed below—with new ones from the literature or field studies, unless those new values are very carefully selected from reliable information compatible with inSTREAM's assumptions and temporal and spatial scales.

Biologists often assume inherent differences among species and that therefore modeling a different species requires re-evaluation of all the trout parameters, and it is easy to find literature suggesting different values for of them. However, our experience trying to extract valid parameter values from the literature suggests a conservative approach to this issue. Rigorous examination of many studies makes it clear that differences in observed rates and potential parameter values that *could* be interpreted as differences among species are just as likely due to differences in study methods or characteristics of the study fish. Further, differences among trout species or populations that could be interpreted as inherent (e.g., size at spawning, tendencies to be nocturnal) can instead emerge in inSTREAM from differences in habitat only. Therefore, our approach is to use the simplest assumption, that differences in trout parameters among species and populations are negligible, absent clear evidence to the contrary.

The following trout parameters do deserve examination and, potentially, new values for new species or sites. Many of these parameters depend on the size and age typically attained by adults, so especially deserve attention when applying inSTREAM to large rivers where trout typically mature and spawn only when relatively old and large. Again, we suggest using the detailed information and literature on each parameter in Sect. 9 in selecting values.

Trout-move-radius-max and ***trout-move-radius-L9*** (Sect. 9.13.1): The distance over which trout explore and select habitat, and how that distance varies with trout size, can depend on site-specific conditions such as stream gradient and complexity. These parameters also strongly affect execution speed; the value of ***trout-move-radius-max*** can be reduced to reduce model execution time.

Trout-fitness-length (Sect. 9.13.2): The value of this parameter should reflect the length of large adult trout, which varies among sites.

Trout-spawn-start-day and ***trout-spawn-end-day*** (Sect. 9.27.3): The timing of spawning certainly differs between fall- and spring-spawning species and populations, and potentially could vary among locally adapted population of the same species.

Trout-spawn-min-age and ***trout-spawn-min-length*** (Sect. 9.27.1): The age and size of maturity can vary substantially within trout species, especially between small and large rivers and across gradients in productivity such as those across latitudes and altitudes.

Trout-spawn-min-temp and ***trout-spawn-max-temp*** (Sect. 9.27.4): These temperature thresholds for spawning must be expected to vary among species and populations that spawn in different seasons or that inhabit different temperature regimes.

Trout-spawn-prob (Sect. 9.27): While this parameter cannot be measured, we can reasonably expect it to be higher where all the criteria for spawning are met on fewer days per year.

Trout-weight-A and ***trout-weight-B*** (Sect. 9.20): These parameters can often be derived from site-specific fish data, e.g., via logarithmic regression on data from high-condition individuals.

Mort-high-temp-T1 and ***mort-high-temp-T9*** (Sect. 9.15): We cautiously recommend reviewing these parameters for differences among species. Unfortunately, the extensive literature on temperature effects on salmonids offers very little that can be reliably interpreted into survival parameter values. Differences in experimental methods and endpoints, individual variability (which could be natural but also result from failure to control variables such as size and energy reserves), incompatible time scales (e.g., survival for 15 days is not comparable to daily survival probability), and other problems make it difficult to identify and quantify differences among species and populations.

Mort-terr-pred-L1 and ***mort-terr-pred-L9*** (Sect. 9.18.1), ***mort-terr-pred-D1*** and ***mort-terr-pred-D9*** (Sect. 9.18.2), ***mort-terr-pred-H1*** and ***mort-terr-pred-H9*** (Sect. 9.18.5), ***mort-terr-pred-hiding-factor*** (9.18.6): These parameters that control how survival of terrestrial predation varies with characteristics of trout and their habitat cells should be revised to reflect the kinds of predators and habitat at each study site. Especially important to consider is the prevalence of predators (e.g., otters) that can be effective in deep and fast water and can extract trout from hiding places. (Sect. 9.18)

Mort-fish-pred-L1 and ***mort-fish-pred-L9*** (Sect. 9.19.1), ***mort-fish-pred-P1*** and ***mort-fish-pred-P9*** (Sect. 9.19.5), ***mort-fish-pred-T1*** and ***mort-fish-pred-T9*** (Sect. 9.19.6): Parameters for survival of fish predation are especially dependent on whether piscivorous fish other than trout are present and the characteristics of such piscivores (Sect. 9.19).

Mort-fish-pred-hiding-factor (Sect. 9.19.4): This parameter should reflect the relative availability of habitat (especially substrate with voids) providing hiding cover for small but not large trout.

Mort-redd-hi-temp-T1 and ***mort-redd-hi-temp-T9*** (Sect. 9.33), ***mort-redd-lo-temp-T1*** and ***mort-redd-lo-temp-T9*** (Sect. 9.9.32): These parameters defining temperature mortality of eggs are expected to vary among species and populations adapted to winter vs. spring incubation.

Mort-redd-scour-depth (Sect. 9.35), ***redd-area*** (Sect. 9.36): These parameters should increase with the size of spawners.

Redd-devel-A, ***redd-devel-B***, ***redd-devel-C*** (Sect. 9.37): Egg incubation rates clearly differ among species.

Trout-depth-suitability and ***trout-velocity-suitability*** (Sect. 9.29): These criteria for spawning suitability likely vary with spawner size; site-specific criteria may be available from PHABSIM studies.

Trout-superind-max-rep (Sect. 9.39): It is appropriate for this parameter to be large (~20-40) for large reaches that use large cells to represent the stream margins often inhabited by small trout, and the computational advantage of large values is especially important for large sites. Smaller values (5-10) make sense at smaller sites, to capture sufficient individual variability and avoid artifacts of smaller cell sizes.

24 Calibration

24.1 Calibration purposes and general considerations

Calibration of a model refers to adjusting parameter values to improve how well model results reproduce observed data. Mechanistic IBMs like inSTREAM differ from traditional simple ecological models in the importance, purposes, and methods of calibration. For simpler models, calibration is often considered the most important step in modeling. The closeness of the calibrated fit of model results to data is often considered the most (or only) important measure of a model's accuracy and usefulness. Calibration is relatively simple because simple models have few parameters and typically produce only one kind of output (e.g., annual abundance). An extensive literature addresses calibration of simple models to data (e.g., Hilborn and Mangel 1997).

For IBMs, however, traditional calibration to data is less important and more complex. Calibration is less important than developing confidence that the model's mechanisms and processes are reasonably accurate: for a complex IBM, calibration is not meaningful unless preceded by analysis and testing of the model's mechanisms and processes (as we do for individual submodels in Sect. 9; see also Sect. 27). In fact, our experience with inSTREAM confirms that well-tested mechanistic models can be used for many kinds of analyses without site-specific calibration. (In Sect. 26.5 we consider how study conclusions based on inSTREAM can be robust to variation in parameter values.) But calibration of complex IBMs is more complicated because they have many parameters and produce multiple kinds of results that each can respond differently to parameter values.

Calibration of inSTREAM to a particular application typically has two purposes: evaluating and improving how well the model reproduces observations, and estimating values of especially uncertain but important parameters. Showing that inSTREAM can reproduce observed data from an application site is not always possible or necessary for the model to be useful. Because many experiments that have already demonstrated the usefulness of its mechanisms (e.g., sects. 3.2, 27), inSTREAM can be useful as a research or decision-support tool even when calibration data are unavailable. (In fact, models that require long-term, site-specific calibration data are rarely useful for management decision support because such data are rarely available.) However, resource managers and reviewers consider a model's ability to reproduce observed data a critical measure of the model's reliability. Therefore, many users of inSTREAM will need to conduct and document some kind of calibration that shows that the model produces plausible results. The second calibration purpose of estimating values of uncertain parameters is almost always relevant because inSTREAM has a very small number of important parameters that are best estimated via calibration (Sect. 24.3).

Both of these purposes can usually be met via a modest calibration process of adjusting the values of 2-4 parameters to make inSTREAM reproduce observed, or at least plausible, values of key results such as adult trout abundance and size.

Railsback and Grimm (2019; their chapter 20) address many issues in calibration of IBMs and provide guidance on methods. Here, we summarize approaches that are particularly relevant to inSTREAM. Calibration typically involves identifying "targets"—criteria defining when the model is adequately calibrated, selecting the parameters to adjust, and then conducting and analyzing simulation experiments to find parameter values that meet the targets. These steps are considered in the following subsections.

24.2 Calibration targets

The first step in calibration is usually to identify which model results will be compared to which observations, and how the comparison will be made. We refer to this step as defining the calibration targets, which are the specific measures of model fit to observations. Developing calibration targets typically includes the following three substeps.

24.2.1 Selecting population measures

Developing calibration targets requires identifying the specific measures of the population that will be compared between model and field observations. Typical measures include trout abundance and mean length, broken out by age classes (e.g., age 1 and age 2 and older). However, the target must also specify the date or time of year of these measures, because trout population measures change over time. Usually, the availability of field data constrains the choice of population measures.

InSTREAM 7's representation of the full daily light cycle adds several potential dimensions to the selection of calibration targets. First, the percentage of fish feeding at different times of day could be treated as a calibration target, even if very coarsely (e.g., a target that not all adult trout feed at night during September, when daytime feeding was observed in the field). In fact, our experience indicates that attempting to closely calibrate the time of day when trout feed could be challenging and not extremely important, even if reliable observations are available (Railsback et al. 2020). Second, the possibility that not all trout are active in daytime is relevant to the development of abundance targets from observations such as snorkel surveys that likely miss fish that are not active in daytime.

24.2.2 Representing time: constant vs. time-series targets

Developing calibration targets requires deciding how they represent time. Calibration targets can be either constant or time-series: inSTREAM can be calibrated to reproduce single numbers (e.g., the long-term mean lengths of age 1 and 2 trout) or time series of observations (e.g., the adult trout abundance observed each October of a five-year period).

Time-series calibration is possible only if data from several years are available. It is valuable for evaluating how well inSTREAM reproduces long-term population dynamics and how they depend on inputs that vary among years. We do not recommend attempting to calibrate inSTREAM against observations at time intervals smaller than one year. InSTREAM ignores or strongly simplifies some seasonal processes, such as variation in food production, so calibration at an annual time scale can be more meaningful and successful than attempting to match observations over shorter times.

Constant calibration targets can be used when no time series of observations are available, or when field observations are not comparable to inSTREAM results because of strong effects of events or processes not represented in the model. Constant targets can be used when no field observations at all are available. Constant targets can be specified as "best" values (e.g., October 1 mean length of age 1 trout as close as possible to 12 cm) or as ranges of acceptable values (e.g., mean age 1 length between 10 and 14 cm).

24.2.3 Comparing model results to observations

Developing calibration targets also requires specifying exactly how inSTREAM's results will be compared to observations and how "best" or acceptable model results will be identified. How to combine or trade off results for multiple population measures is an important consideration, because one set of parameter values rarely produces the best calibration for multiple calibration targets such as the abundance and size of several age classes.

In time-series calibration, model results can be compared to observations using one of two closely related methods. One is to rank results from each alternative parameterization (discussed below) by the sum of squared differences between model-predicted and observed values; smaller values of this measure indicate better model fit. The second method is to regress model-predicted and observed values against each other on an X-Y plot and evaluate alternative parameterizations by their correlation coefficient, how close the slope is to 1.0, and how close the Y intercept is to zero (Smith and Rose 1995). Using such measures of model fit becomes ambiguous and requires judgement when attempting to calibrate more than one kind of model result, e.g., by considering both length and abundance of several different age classes. We can, for example, evaluate model fit for each model result separately and look for parameter values that provide relatively good fit for the most kinds of results.

Constant targets have the advantage of being clearer to apply to multiple kinds of model result. One approach is to look not for parameter values that provide the "best" fit for any model result but instead for values that provide acceptable fits for all results. We can, for example, specify ranges of acceptable values for four kinds of result: October 1 length and abundance of age 1 and age 2 and older trout, averaged over five simulated years. Calibration can then evaluate parameter combinations by how many of these four kinds of result fall within their target ranges. This approach is similar to "parameter filtering" or "pattern-oriented parameterization" (Grimm and Railsback 2005, 2019), in which a variety of observed patterns or ranges in different kinds of results are used as "filters" to identify good combinations of parameter values.

These approaches can be combined. For example, we can first "filter" all the parameter combinations examined to exclude those that produced trout lengths outside a target range, and then use the regression approach on the remaining combinations to find the one best fitting a time series of abundance.

24.3 Calibration parameters and outputs

The second major calibration step is selecting the inSTREAM parameters to adjust. In doing so, we can use one lesson from traditional, simple ecological models: calibrating fewer parameters increases confidence in their calibrated values. Trying to adjust too many parameters reduces our ability to find the best range of values for each, so adjusting as few parameters as possible is generally recommended. The parameters most suitable for calibration are those to which model results are highly sensitive (Sect. 26.2) and for which there is little other basis for selecting values. These considerations, plus our experience with inSTREAM, identify five parameters that are especially suited for calibration. We do not recommend calibrating *all* these parameters at once; in fact, we typically use only the first 2-3 of them at any particular site. However, the choice of which of these parameters to calibrate can depend on what data are available and which outputs most need adjustment. For example, juvenile abundance and size are typically known with less certainty and are of less management importance, so we focus calibration on parameters that most strongly affect adult results.

We recommend considering the following five parameters for calibration. We illustrate their typical effects on model results via sensitivity analyses that varied the calibration parameter over a wide range and examined how simulated trout abundance and length responded (Figure 68). We examined two age classes: age 1, and 2 and older. The experiments represented a hypothetical trout population in the "RESTORED" stream reach of Railsback et al. (2021).

- Concentration of drift food (***reach-drift-conc***): This parameter strongly affects growth and survival of all age classes (Figure 68A). Data on actual drift availability are rarely available and, when available, typically highly variable and uncertain. (We do not expect calibrated values of ***reach-drift-conc*** to match observed drift concentrations because this parameter also captures effects of several processes that are neglected in inSTREAM, e.g., variation in prey size and energy content, and spatial and seasonal variation in drift production.) Drift food concentration is typically the first parameter to consider in calibration.

- Survival of terrestrial predation (***reach-terr-pred-min***): Terrestrial predation is typically a dominant source of mortality for all but the smallest trout, so this parameter strongly affects trout abundance (but not size) and actual predation rates are almost never known (Figure 68B). This parameter is particularly useful for adjusting abundance with little effect on size.

- Regeneration distance for drift food (***reach-drift-regen-distance***): This parameter affects the total availability of drift food per cell, which can limit how many trout can occupy each high-quality cell, but not the drift intake and growth of the individual trout. Consequently, it can strong affect trout abundance without affecting size (Figure 68C). This parameter could be calibrated by attempting to match observed densities of trout in high-quality habitat.

- Production of stationary food (***reach-search-prod***): Search feeding typically is profitable only for some juveniles throughout the day and for adult trout feeding at night, although such fish often also use drift feeding. Therefore, the parameter typically affects size of age 0 and 1 trout, with little effect on older fish (Figure 68D). This search food parameter is therefore useful for calibrating differences in size between earlier (ages 0 and 1) and later age classes.

- Survival of fish predation (***reach-fish-pred-min***): Small juvenile trout are highly vulnerable to predation by other fish, but the actual risk is highly variable and difficult to quantify. Therefore, this parameter can be useful for calibrating the abundance of age 0 and, to a lesser extent, age 1 trout (Figure 68E-F), even if it has little effect on adult abundance. This parameter can have strong effects on adult abundance when inSTREAM is parameterized to represent predation by non-trout fish (which it is not in Figure 68).

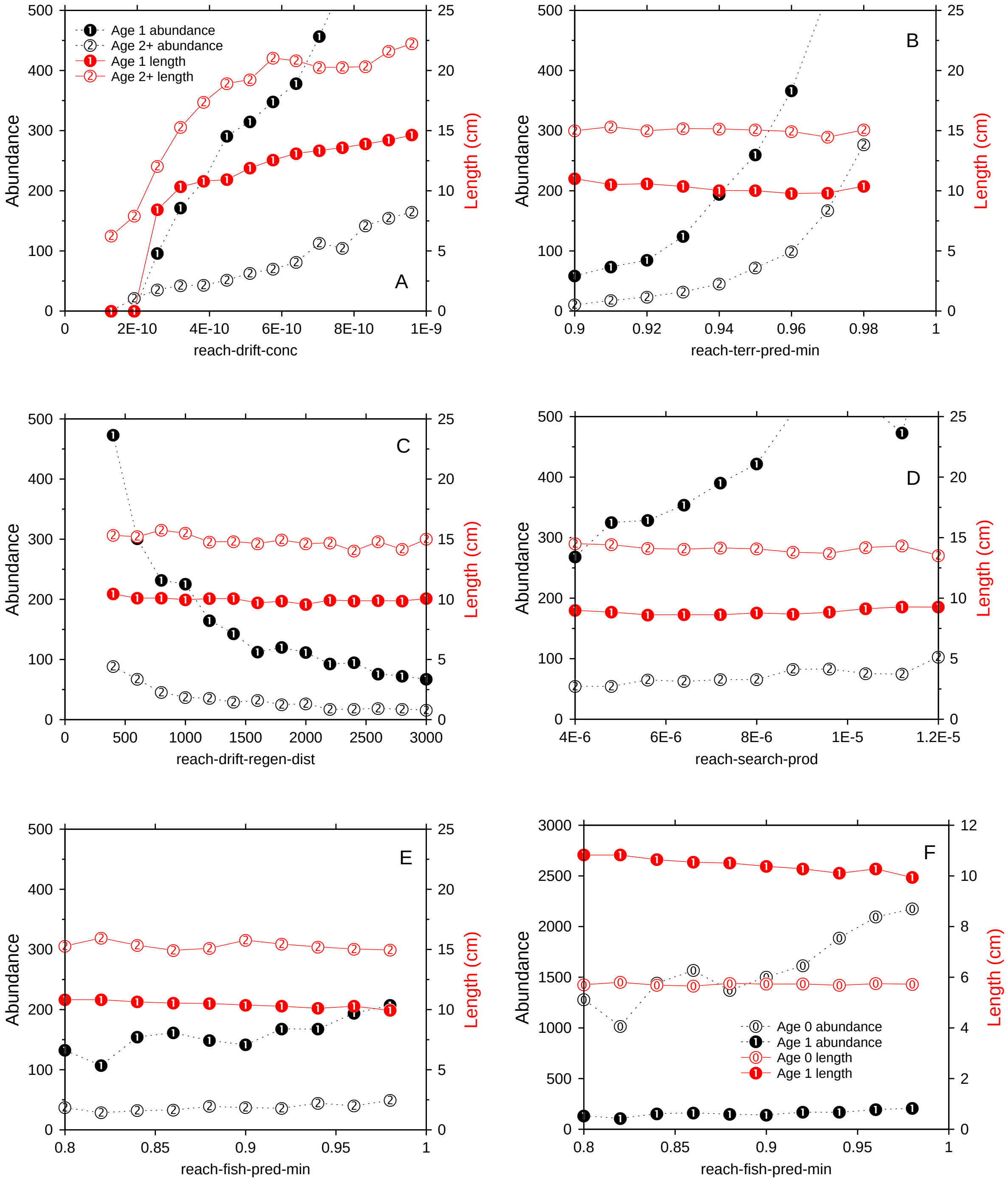


Figure 68. Example sensitivity experiment for parameters typically used to calibrate inSTREAM. Graphs display response of simulated abundance (left Y axis) and mean length (right Y axis) of age 1 and age 2 and older trout to: (A) ***reach-drift-conc***, (B) ***reach-terr-pred-min***, (C) ***reach-drift-regen-distance***, (D) ***reach-search-prod***, and (E) ***reach-fish-pred-min***. Graph F illustrates response of age 0 and 1 trout to ***reach-fish-pred-min***; note its different Y axis scales.

24.4 Calibration experiment designs

The next calibration step is designing the simulation experiments that will produce output to be compared to observations. Here we discuss two relatively simple and direct ways of conducting simulation experiments to find useful values of calibration parameters.

[We do not provide guidance on the more sophisticated and comprehensive approaches that the simulation modeling literature and software tools for NetLogo make possible for users with the necessary computer resources. The NetLogo software platform used by inSTREAM 7 includes a parameter-fitting tool called BehaviorSearch that provides a third approach. BehaviorSearch uses sophisticated optimization methods to find parameter values that cause a NetLogo model to fit a specific numerical criterion. However, BehaviorSearch is unlikely to be a good calibration approach for inSTREAM because it requires a single measure of model fit—discussed to some extent in Sect. 24.2.3—and depends on larger numbers of model runs than are often feasible with inSTREAM. It is also not clear whether inSTREAM's stochasticity would prevent the optimization methods from reaching a solution. Salecker et al. (2019) provide similar capabilities via a package in the R statistical language for automated analysis of NetLogo models. We have not yet attempted to calibrate inSTREAM using either of these tools, but Ayllón et al. 2016 used a comprehensive "Latin hypercube" approach for calibrating an earlier version of inSTREAM that was also in NetLogo.]

One parameter at a time. The first approach fits inSTREAM's results to observed data by adjusting one parameter at a time. Each parameter can be calibrated simply by trying several different values and trying to zero in on a good value, but it is typically more efficient and rigorous to use a sensitivity experiment approach (Sect. 25.3.2; Figure 68), running the model several times using a wide range of values. The sensitivity experiment approach illustrates how the model responds to the calibration parameter and how much of the variation in results is due to stochasticity. Good parameter values can be interpolated from the sensitivity response.

A coarse (but often adequate) calibration of inSTREAM can often be achieved using the one-at-a-time approach. First, fit the size (length or weight) of age 1 and older trout by adjusting ***reach-drift-conc***. Then fit the abundance of age 1 and older trout by adjusting ***reach-terr-pred-min***. If reliable calibration targets are available for age 0 trout, their abundance can be fit by adjusting ***reach-fish-pred-min*** and their size by adjusting ***reach-search-prod***. After each of these steps is completed, of course it is necessary to check whether the previous steps have been affected. For example, adjusting ***reach-search-prod*** is likely to have some effect on adult size, but perhaps not enough to require re-calibration of ***reach-drift-conc***.

Parameter combinations. The second approach uses simulation experiments that vary more than one parameter at a time to "explore the parameter space" and systematically search for good combinations of parameter values. This approach can be implemented by selecting ~4 values, over wide ranges, of two or three parameters (usually, ***reach-drift-conc*** and ***reach-terr-pred-min***), and then running inSTREAM with all combinations of the parameter values and identifying combinations that best meet the calibration targets. If one or several measures of model fit to data can be developed (e.g., the sum of squared differences measure discussed in Sect. 24.2, applied separately to adult trout length and abundance), then contour plots can be used to observe how the value of those measures responded to the calibration parameter values in the simulation experiments, and to interpolate good parameter combinations. Examining such contour plots of fit to (e.g.) adult abundance and length allows identification of parameter combinations that best meet both targets. Additional simulation experiments over narrower ranges of parameter values can then refine the search for good combinations.

24.5 Calibration model run setup

The final step in designing calibration methods is selecting the model input and settings. Here we address several important considerations in how to set up the simulation experiments used for calibration: what initial conditions to use, the duration of model runs, and use of replication. NetLogo's BehaviorSpace tool (Sect. 15; not to be confused with BehaviorSearch) is invaluable for automating calibration experiments, reducing the time for both setting up and executing them; we strongly recommend its use.

Initial conditions refers to the input and parameters that define the virtual trout population at the start of a model run (Sect. 7). Ideally, initial conditions are unimportant because the calibration simulations will be long enough to eliminate the effects of initial population characteristics (discussed below). We typically initialize calibration simulations with populations resembling the calibration targets: with approximately the same number and size of trout, by species and age class, that we are calibrating the model to predict. This approach should make model results even less affected by initial conditions than would an initial population very different from the calibration targets.

Model run duration is determined by two factors: how much time the calibration targets represent and how long inSTREAM needs to run to make initial conditions unimportant. When using time-series calibration (Sect. 24.2.2) we of course need to simulate the same years represented by the field observations. When using constant calibration targets, model runs need to be long enough to adequately represent the range of conditions that the targets are assumed to represent. This duration can depend on the variability of conditions at the study site. For example, where flow and temperature regimes are closely constrained by a reservoir, relatively short simulations can be adequate.

We address the effects of initial conditions on model results in Sect. 26.3, which indicates that 3-5 years can be assumed sufficient to make inSTREAM results reasonably unaffected by initial conditions. Therefore, we typically exclude the first 3-5 years of simulations from the analyses. If input for such "warm-up" years is not available, it can be generated by copying and repeating the input for subsequent years.

Replication is a consideration for all simulation experiments (discussed further in Sect. 25): do we need to run each set of parameter values several times to distinguish its effects from those of stochasticity?

In general, we find it more useful to simulate more parameter values than to replicate each value, perhaps with the exception of when calibrating a small and therefore particularly stochastic virtual population. For example, if it is computationally feasible to complete ~130 model runs for a calibration experiment, it is typically preferable to use all combinations of five values of three calibration parameters instead of five replicates of three values of each parameter. However, the calibration analyses should consider that some variation in model results is stochastic: parameter combinations that seem to produce especially good fit to observations deserve further examination (e.g., with a second experiment over narrower parameter ranges) to determine the influence of stochasticity.

25 Simulation Experiments

25.1 General study design considerations

Models like inSTREAM are applied to management or research problems by designing and conducting simulation experiments: controlled experiments conducted on the model, just as scientists conduct controlled studies in the field or laboratory. In many ways designing simulation experiments is similar to designing field studies—the study design must determine:

- Which inputs (data or parameters) to "control" (hold constant) and which to vary in what ways,
- What model output must be "observed" from the experiment,
- How many replicates are needed, and
- What statistical or graphical methods will be used to interpret the results.

One key difference between simulation experiments and field studies is that simulation experiments are cheap, fast, well-controlled, and highly observable. Instead of relying on extensive statistical analysis of relatively few data (the typical situation with field studies) we can run more simulation experiments. Grimm and Railsback (2005; see their Chapter 9) and Railsback and Grimm (2019) provide general guidance on study design and analysis of IBMs; here we focus on issues specific to inSTREAM.

Two terms (also defined in Sect. 2.1) are important in discussing study design. A *scenario* is a complete set of input to inSTREAM that represents one particular set of environmental conditions and one management alternative. Effects of various environmental conditions (e.g., wet, average, or dry flow years) or management alternatives (e.g., instream flow requirements A, B, or C) are typically assessed by comparing output produced by several different scenarios. *Replicates* (or "repetitions" in NetLogo terminology; Sect. 15.2) are multiple models runs that represent the same scenario (use exactly the same input data and parameters) but use different pseudo-random numbers; variation in results among replicates represents the effects of the model's stochastic processes. (We also consider another replication method in Sect. 25.3.1.)

Some issues to consider in designing inSTREAM experiments are:

- The model produces many kinds of output, including time series of population abundance and biomass broken out by age and species, mortality rates for each mortality source, habitat availability and use, the number and timing of spawning events, and how many eggs died of each mortality source and how many survived to emergence. Simulations of multiple species can predict the relative abundance of each species, and long-term simulations can predict the frequency of local extinction under various conditions.
- Model results are stochastic, mainly because spawning success and trout mortality are stochastic. Therefore, it is important to understand the extent to which differences in model results for different scenarios may be due to chance.

- Model assumptions, parameters, and input data are uncertain, so model predictions are uncertain. These uncertainties need to be acknowledged and addressed when using the model to support decisions. (This topic is addressed in Sect. 26.5.)
- Natural variability in physical habitat (river channel shape, etc.) and in flow and weather can have strong effects on results, sometimes stronger than those of the management alternatives the study is designed to compare. The effects of natural variability is another kind of uncertainty in model results.
- While simulation experiments using inSTREAM are far faster and cheaper than field studies, there are limits to how many model runs are feasible.
- The arbitrariness of conventional statistical methods for comparing scenarios (Hilborn 1997) becomes especially apparent with data generated by a simulation model. Whether the results of two scenarios are "significantly different" depends on the number of replicates and the magnitude of differences among scenario inputs, which are both easily manipulated in simulation experiments (Wilson et al. 2014).

The importance of each of these issues depends on the study site and the problem addressed. The following subsections provide general guidance based on our experience with inSTREAM.

25.2 Outputs to analyze

One of the first questions in designing a study using inSTREAM is which of the model's outputs to analyze. It is completely impractical and counterproductive to analyze *all* the model's possible results: attempting to do so is more likely to paralyze than support decision-making. However, inSTREAM's many kinds of output allow us to do what we cannot with field studies: look at the primary results and then explore further to learn what caused them. The model is far more valuable when we do not just look at its key outputs but instead then use it as a tool for understanding how different processes affect simulated, and therefore perhaps real, trout populations. Analysis of results should therefore be an adaptive process in which we first look at primary outputs and then selectively examine others.

Effective analyses often start by examining only one or two key outputs. Focusing on a few obviously important results usually provides clearer, less-confusing support for management decisions. Especially for management decision support such as instream flow assessment, study design typically focuses on the abundance and size of adult trout. In fact, population biomass (total weight of all trout, or all adult trout) is a useful measure that reflects both abundance and size of individuals. Abundance and size are clear, measurable indicators of the status of fish populations; managers and the public can easily relate to these indicators, and comparable measures of real populations are often available.

Of course other study goals may require focus on other outputs. If the viability of a threatened trout population is the key management interest, then simulated population persistence may be the most important model output. Persistence can be evaluated as, for example, the number of replicate 50-year simulations in which extinction occurred, out of 50 total replicates. Relative abundance of native vs. exotic trout species can be an important output when management goals include promoting native species.

Despite the importance of focusing on a few key outputs, analyzing *only* those outputs is completely inadequate. Conclusions, recommendations, and management decisions will gain value from analyses that examine more detailed outputs to understand primary outputs such as abundance and biomass.

Examining why inSTREAM produced the results it did is essential for several reasons. First, inSTREAM often produces results strongly driven by mechanisms that, upon inspection, are not very credible and need modification to produce more believable results. A typical example is a set of instream flow scenarios that produce sharply lower abundance above a certain flow. That result could arise from mechanisms as simple as the higher instream flows often exceeding the maximum flow at which trout can spawn (***reach-max-spawn-flow***; Sect. 9.27.5) or spawning-season temperatures rarely exceeding the lower threshold for spawning (***trout-spawn-min-temp***; Sect. 9.27.4). Because real trout could likely adapt to such changes by spawning at different times or flows, prudent users would try parameter values that let trout spawn under all scenarios and repeat the simulation experiment.

On the other hand, determining what mechanisms drive primary population responses often leads to valuable insight on what strongly affects the simulated trout population and, therefore, how the real population might be managed more successfully or cost-effectively. InSTREAM very often identifies important mechanisms that biologists would otherwise be unlikely to think about; examples include:

- High mortality of redds due to temperature, scour, or superimposition;
- Changes in temperature regime during incubation that, by changing egg development rates, cause more (or fewer) fry to emerge when flow is high and fry habitat scarce;
- Conditions that reduce adult trout growth (e.g., low flows, high temperatures) before the spawning season and, therefore, the number and fecundity of spawners;
- Low-growth conditions after spawning that reduce the post-spawning survival of adults; and
- Changes in temperature regime during high-growth seasons (spring, fall) that affect trout fecundity and ability to survive subsequent seasons.

Uncovering such mechanisms requires creative detective work using methods that must be adapted to each situation (see also Sect. 25.4 below). We recommend users start by examining outputs such as:

- Time series of abundance, by age class. Where in the life cycle did abundance diverge among the scenarios?
- Time series of length, by age class. Are there major differences among scenarios in how much growth occurs when?
- What percentage of adults spawned? If few adults spawned, why? Which of the spawning criteria (Sect. 9.27) were not met?
- What percentage of eggs died, of what causes?

(The detective work to identify important mechanisms can include not only analysis of additional outputs but also the use of new, often unrealistically simplified, simulation scenarios as discussed in Sect. 25.4. These scenarios can be designed to test specific hypotheses about what mechanisms in the model explain key results.)

Trout mortality results (how many simulated trout died of which causes) are often surprisingly unhelpful for understanding simulation results. The reason is that the habitat and activity selection behavior produces tradeoffs among mortality risks that obscure the importance of different stressors. For example, a scenario with high summer temperatures increases trout metabolic rates; consequently, the trout decide to feed more often and in riskier conditions (more often in daytime or in less-safe habitat) to get the food they need to reduce the risk of starvation. As a result, the effects of high temperatures show up as increases in mortality due to predation and starvation as well as high temperature. We cannot infer what stressors most affect the population by looking only at what trout died of. Another reason mortality output can be misleading is that high numbers of dead trout automatically result from high abundance: all trout die eventually, so the scenario that produces the highest number of dead trout is also the scenario that produced highest abundance.

25.3 General experimental designs

This section briefly describes two common general designs for studies using inSTREAM: scenario comparisons and sensitivity experiments. Scenario comparisons resemble the study designs commonly used in field or laboratory studies, but sensitivity experiments often can provide more information about how and why fish populations are predicted to vary.

25.3.1 Scenario comparisons

Scenario comparisons parallel traditional field studies: several distinct scenarios are defined, simulated, and compared. This approach is natural when the study purpose is comparing distinct management actions: assessing several alternative rules for instream flow releases, examining the effects of introducing another trout species, or deciding which of several channel restoration projects to implement.

The first step in conducting scenario comparisons is to define the scenarios, by specifying what model inputs (parameter values and data such as channel shape, flow, and temperature; and, possibly, how many and which species) will vary among scenarios. For example, if the objective is to compare three sets of alternative instream flow release rules for a reservoir, then defining the scenarios could require generating (e.g., with a reservoir water balance model) the time series of daily flows that would result from each alternative rule set. If flow strongly affects water temperature, defining the scenarios could also require generating a time series of daily temperatures corresponding to each daily flow time series. In another example, Harvey and Railsback (2009) compared five turbidity scenarios, each differing only in the assumed relation between turbidity and streamflow. One scenario represented an undisturbed watershed in which turbidity increased slowly with flow (e.g., 25 NTUs turbidity at flow of 1.0 m^3/s), another scenario represented a highly disturbed watershed with rapid increases in turbidity with flow (e.g., 190 NTUs at 1.0 m^3/s), and three represented intermediate flow-turbidity relationships.

The second step in a scenario comparison study is to decide what kind of replication to use. Normally, replication of stochastic simulation models is done by simply altering the random number generator seed to evaluate variability in results due to the model’s stochastic processes. This type of replication is commonly used in inSTREAM experiments. But inSTREAM’s “year shuffler” facility (Sect. 8.2) provides an alternative replication method: randomly re-ordering the years of input data. Simulations replicated by randomizing the input data years examine how natural variability in flow, temperature, and turbidity (in addition to the model's stochastic processes) affect results.

Next, users must decide on the appropriate level of replication. Using too few replicates makes it impossible to understand how much of the variation among scenarios is due only to stochasticity, but using too many replicates is computationally burdensome and can produce statistical significance among scenario results when their biological significance (or likelihood of being detected in the field) is low. Using around 5-10 replicates has appeared sufficient to identify differences among scenarios that seem likely to be important for management. Five to 10 replicates are insufficient to precisely define the distribution of stochastic results for a scenario, but if two scenarios cannot be distinguished by this many replicates, then their difference seems unlikely to be important compared to the effects of natural variability and model uncertainty.

Results may be analyzed statistically or simply displayed. Statistical comparisons (typically, using two-sided t-tests or analyses of variance) can be informative, especially for audiences accustomed to statistical analysis. Results of replicated scenarios are likely to meet the assumption of normality for parametric statistics (except when "extinction" is common). However, simple graphical comparison of results (e.g., via bar charts with error bars, or scatter plots showing the value of each replicate—see Magnusson 2000) are often just as informative and easier to interpret. For examples of both statistical and graphical comparison of scenario results, see Railsback and Harvey (2002) and Railsback et al. (2005).

25.3.2 Sensitivity experiments

Sensitivity experiments are experimental designs intended to provide more and broader information about population responses to management actions. Instead of comparing a few discrete scenarios, a sensitivity experiment provides an "incremental" analysis of predicted trout populations responses to a range of actions. For example, an experiment could look at the population's sensitivity to summer instream flow by simulating a broad range of instream flow scenarios. (The parameter sensitivity analyses in Sect. 26.2 are one kind of sensitivity experiment. Here, we look at model sensitivity to inputs other than parameter values; Sect. 26.3 provides an example.)

Version 5 of inSTREAM included a "Limiting Factors Tool," software for setting up and executing sensitivity experiments for flow; summer and winter temperatures; food availability; availability of spawning gravel, velocity shelter, and hiding cover; predation risk; and redd scour. Fish biologists often think of a limiting factor as one such process that has a dominant effect on population abundance. In inSTREAM, because of its tradeoff behaviors, we never see just one or two such dominant factors, but it is still useful to determine how strong the effects of each of these factors are at a particular study site. We can do this with sensitivity experiments. InSTREAM 7 does not currently include a built-in tool for limiting factors analyses, in part because it is easy to set them up using the methods described here and BehaviorSpace (Sect. 15).

The general steps in a sensitivity experiment are similar to those for a scenario comparison. First is to identify and define the scenarios to be simulated. Typical experiments use scenarios that vary only one or two inputs across broad ranges. An incremental analysis of instream flows would include perhaps 10-20 flow scenarios from very low to very high flow (e.g., Railsback et al. 2021). As another example, a sensitivity experiment on the effects of weekend pulse flow releases for whitewater recreation could include scenarios ranging from zero to 20 weekends of whitewater release per year. Multivariate sensitivity experiments are also possible, for example by defining 50 scenarios that include 10 levels of minimum instream flow and 5 levels of whitewater release. It is useful to extend the simulated scenarios beyond the range considered feasible for management, to understand trends even at their extremes.

The second step is to execute the scenarios. Typically, only one replicate is executed per scenario because it is not as important to understand the extent of variability in results for each scenario: output from many scenarios illustrates both the model's response to the input being varied and the noise in this response.

Using only one replicate of many scenarios does not mean that stochasticity is ignored in analyzing results. Analysis of results in a sensitivity experiment can use graphical and statistical techniques, but the techniques differ from those for scenario comparisons. The first analysis step should always be to plot how the trout population output of inSTREAM varied over the range of scenarios. Did the population increase or decrease consistently over the range? How "noisy" is the response? Stochasticity typically shows up as random variation around a smooth response. Experiments sometimes produce little or no detectable trend, indicating that the "signal" from the variable that changed among scenarios is small compared to the stochastic "noise" in the results. For multivariate sensitivity experiments, contour plots can be useful to examine responses.

25.4 Variation in inputs: Realistic vs. unrealistic scenarios

Another basic study design decision addresses how much realistic variation to include in the input scenarios. One advantage of IBMs such as inSTREAM is that they can naturally predict the effects of realistic, day-to-day variation in the driving inputs: flow, temperature, and turbidity. These variables typically vary from day to day as well as seasonally, even downstream of major reservoirs. However, including more variation in a simulation experiment can make it more difficult to thoroughly understand its results or distinguish among scenarios. Therefore, designing studies requires choosing between scenarios with realistic levels of variation in the time-series inputs and unrealistic scenarios with variation in some of the inputs suppressed to make analysis easier.

In general, if the purpose of a study is assessment of management actions, there is less interest in understanding the details of how results arose and more interest in just predicting how trout populations respond to the actions. In this case, scenarios with realistic levels of variation may be most appropriate because they compare management actions in a simulated context most closely resembling the real situation. If, however, the purpose of a study is more focused on understanding the processes by which river management affects trout populations, then experiments using scenarios with unrealistically reduced variation of some variables may be most appropriate. In many cases, both approaches are useful: we can start with a more realistic level of variability and then use simplified, unrealistic scenarios as part of the detective work needed to reveal the mechanisms driving the primary results (Sect. 25.2).

25.5 Conclusions and summary guidance

Outlining an appropriate study design should be one of the first steps in applying inSTREAM to either river management or research studies. Not only does the study design influence the kinds of inputs and analyses needed, but the process of developing the study design helps clarify how the modeling work can support the management decisions or research objectives.

The most important concept in designing studies that use inSTREAM is that of simulation experiments: given appropriate input and calibration, the model can be considered a laboratory in which we conduct controlled experiments. Those experiments should be explicitly designed to address the purpose of the inSTREAM application: supporting a management decision, answering a research question, etc.

Many applications of inSTREAM are expected to have the objective of comparing distinct management alternatives, for example alternative instream flow policies that determine daily flow releases. The scenario comparison approach is usually appropriate for such situations: specify sets of model input to represent each of the alternatives and use replicate simulations to examine the degree of predicted difference in trout populations among the alternatives. Model results can be used to rank the alternatives by their predicted benefits to trout (e.g., simulated average annual population biomass). The robustness of the analysis can be examined, using methods presented in Sect. 26.5, by determining how the rankings change when simulations use different values of key parameters, or different weather and hydrologic conditions, or alternative values for other particularly uncertain inputs.

For many studies it will also be useful to conduct sensitivity (or “incremental”) analyses of how trout populations are predicted to vary over a broad range of inputs such as flow or temperature. These analyses are more useful for understanding how management variables (and potential “limiting factors”) affect trout and for finding good management policies. Results can be analyzed by developing graphical (and perhaps also statistical) relationships between measures of the trout population and values of the inputs. Robustness of results can be examined by determining if and how these relationships change across different values of particularly uncertain and important inputs. Especially for sensitivity experiments, users need not be constrained to using input with “realistic” levels of variability; instead, unrealistically limiting variability can help to understand some problems.

For either the scenario comparison or sensitivity experiment approach, the greatest value of inSTREAM is the ability to dig deep into results to understand why they arose and learn what they say about the real system. In some cases we learn that seemingly important model results arise from assumptions, parameter values, or inputs that are questionable and need reconsideration, so the whole analysis cycle needs to be repeated. In other cases we learn what mechanisms appear to drive trout populations and, therefore, what management measures may be cost-effective for protecting or enhancing them.

26 Sensitivity and Uncertainty in InSTREAM 7

26.1 General considerations

This section addresses basic questions about model sensitivity and uncertainty: To what inputs and parameters are inSTREAM results most sensitive? How should these sensitivities affect model use? How do uncertainties in parameter values affect model results? How robust are conclusions drawn from inSTREAM to parameter values and other inputs? More general guidance on sensitivity, uncertainty, and robustness analysis of IBMs is provided by Grimm and Railsback (2005), Railsback and Grimm (2019), and Grimm and Berger (2016).

In the following subsections, we address these questions in part by using example simulation analyses. Unfortunately, all of these analyses produce conclusions specific to the sites and questions we modeled; while we attempt to draw general conclusions, none of the guidance we provide can be considered universally applicable. The sensitivities and uncertainties in inSTREAM results must be expected to vary among sites, especially for sites subject to particularly strong stressors such as extreme temperatures, turbidities, predation pressure, or competition among species. There can even be important variation in sensitivities and uncertainties among scenarios for the same site.

26.2 Sensitivity of primary predictions to parameter values

Here we present a conventional parameter sensitivity analysis of inSTREAM 7. The purpose of the analysis is to quantify the effect of each parameter on the model's primary outputs, its predictions of trout abundance and size. Parameters to which inSTREAM is particularly sensitive deserve special attention in parameterization and calibration and may be useful subjects of empirical research to improve the model.

This analysis closely follows a previous analysis of inSTREAM 4.2 by Cunningham (2007), summarized by Railsback et al. (2009). We used a one-parameter-at-a-time design, varying each parameter by itself with other parameters constant at their standard values. We used the same physical habitat input as Cunningham (2007), the lower mainstem site of Little Jones Creek as described by Railsback and Harvey (2001). We first calibrated inSTREAM 7 to produce realistic values of the abundance and length of age 1 and 2 and older trout at this site. We used observed flows and temperatures for a 10-year period, water years 2000-2010. We analyzed the total biomass (kg) of trout age 1 and higher on September 30th, averaged over the last five of the 10 simulated years.

For each parameter, we estimated a range of realistic values, as a percentage of the standard value. We used wide but not extreme ranges of possible parameter values, usually centered on the standard value. The range for each parameter was based on the extent of empirical evidence for its standard value, and limited to values producing reasonable results (some parameters produce unrealistic results outside of a narrow and well-defined range).

We ran 21 simulations for each parameter, using values across its range. For example, if a parameter was given a range of ±20%, then the 21 simulations used values of 80%, 82%, 84%, ... 120% of the parameter's standard value. Simulations for a parameter with a range of ±50% used values of 50%, 55%, ... 150% of the standard value. For the analysis, the 21 values of each parameter were replaced by "scaled" values of 0.0 to 1.0 in increments of 0.05. (The lowest value of each parameter had a scaled value of 0.0, the middle value had a scaled value of 0.5, and the highest was 1.0.)

We calculated two measures of parameter sensitivity. First was the slope of the population biomass with respect to the scaled parameter values (the "scaled slope"), calculated using linear regression. Second was the "unscaled slope", the slope of population biomass with respect to the parameter value as a percentage of its standard value. The scaled slope represents the biomass response to a parameter over the full range of realistic parameter values, which is especially important for parameters producing nonlinear responses. For parameters producing linear responses, the unscaled slope is less dependent on the range's magnitude.

The effects of a few parameters were clearly nonlinear, with different slopes over different ranges of parameter values. In these cases, we report the slope over the range (at least 5-7 parameter values) where it is steepest.

These sensitivity measures could not be used for two kinds of parameters. Parameters representing dates (e.g., the date at which spawning starts) were instead analyzed by varying the date without scaling it. The sensitivity measures also do not work for parameters with a standard value of zero (because ±*X*% of 0.0 is 0.0); such parameters were either not analyzed or varied in another way that does not produce slopes comparable to those of other parameters.

Results of the sensitivity analysis are presented in separate tables for habitat parameters (Table 32), trout parameters except those for survival (Table 33), trout survival parameters (Table 34), and redd parameters (Table 35). These tables present each parameter's standard value in this application, the range of values simulated, the scaled and unscaled biomass response slope, and a graph of mean age 1+ biomass response to the scaled parameter value. The graphs include a regression line for all 21 parameter values. Notes in the tables indicate parameters with nonlinear responses and those for which the slope method does not work.

Table 36 provides a summary of the analysis, listing the parameters in descending order of sensitivity as the absolute value of scaled slope. This table omits the parameters for which this slope cannot be calculated.

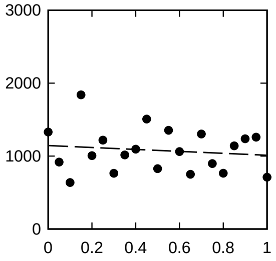
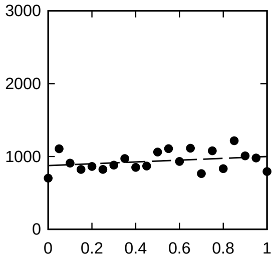
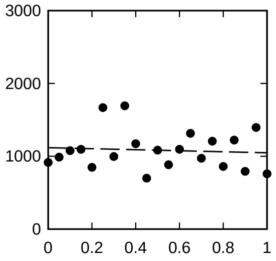
Interpretation of these results must recognize that the study site has consistently benign temperatures and low turbidity, and abundant velocity shelter and hiding cover. Therefore, parameters related to effects of these variables should be expected to have stronger effects at less-benign study sites. These results for Little Jones Creek are useful for identifying parameters likely to have strong effects at all sites, but not for identifying parameters that always have little effect.

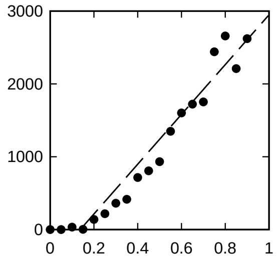
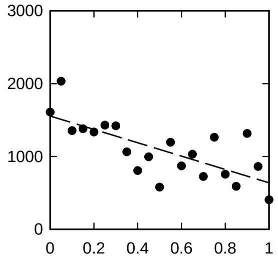
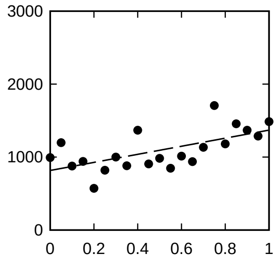
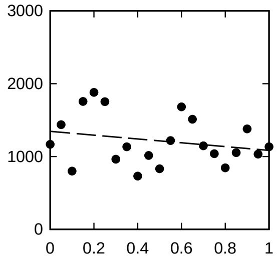
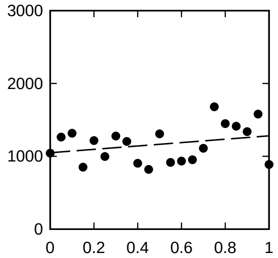
The analysis identified 14 parameters with absolute value of scaled slope greater than 1000, and 26 parameters with scaled slopes greater than 500. The high sensitivity to these parameters indicates that the following processes are especially important in inSTREAM 7, in approximate order of decreasing importance:

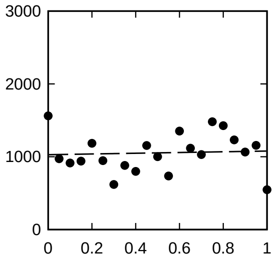
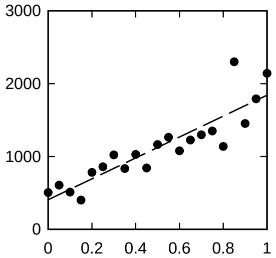
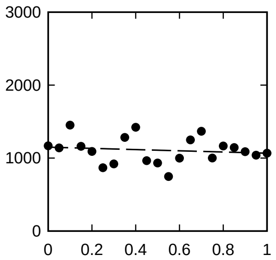
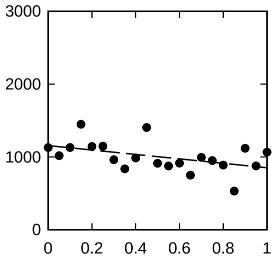
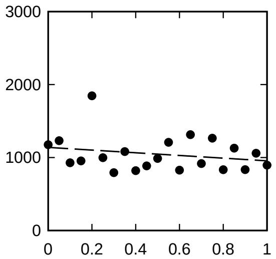
- Drift food availability;
- Terrestrial predation and how it varies with habitat (depth, velocity, escape cover);
- Drift feeding success and how it depends on light, turbidity, velocity, and fish size;
- The habitat selection radius;
- Trout maximum sustainable swimming speed;
- Respiration rates and energy densities; and
- Poor condition survival.

Knowing which parameters have relatively little effect on results is also important. Parameters driving reproductive success generally had relatively small effects. Of particular note are the small effects of the two superindividual parameters (Table 33). Low response to these parameters is reassuring, but users should re-analyze those effects before assuming that large superindividuals can be used with negligible effects at other sites. Superindividual parameter values interact with cell sizes and availability of habitat resources: larger superindividuals are expected to create more simulation artifacts and uncertainty when cells are small and when availability of food and cover for feeding and hiding are relatively low.

Table 32. Sensitivity analysis results for global and reach habitat parameters.

Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
latitude	40	20%	-133	-333	
light-correction	0.7	20%	124	311	
light-at-night	0.9	80%	-70	-44	

reach-drift-conc	5.9E-10	50%	3705 (excluding lowest two values)	3705	
reach-drift-regen-distance	1000	50%	-916	-916	
reach-fish-pred-min	0.8	10%	553	2766	
reach-search-prod	5.0E-07	50%	-262	-262	
reach-shading	0.7	20%	232	580	

reach-shelter-speed-frac	0.3	40%	50	63	
reach-terr-pred-min	0.71	10%	1435	7176	
reach-light-turbid-coef	0.0017	20%	-83	-207	
reach-light-turbid-const	0.0	Excluded because standard value is zero			
max-spawn-flow	4.0	50%	-304	-304	
reach-shear-A	0.019	20%	-184	-461	

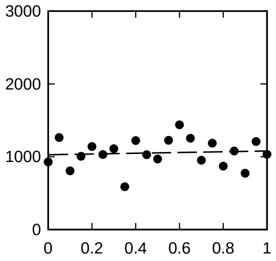
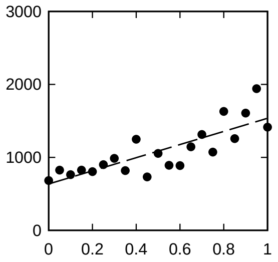
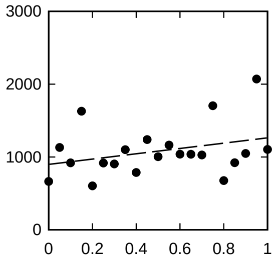
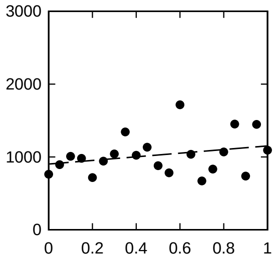
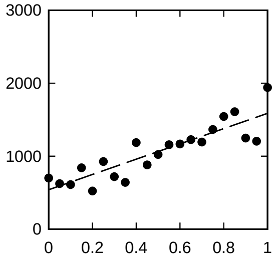
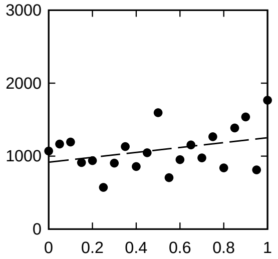
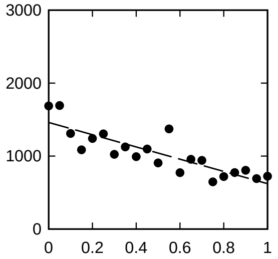
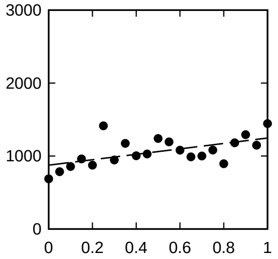
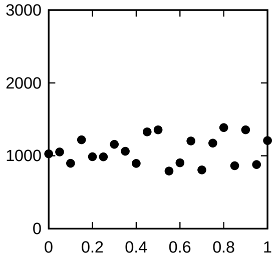
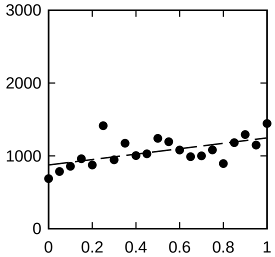
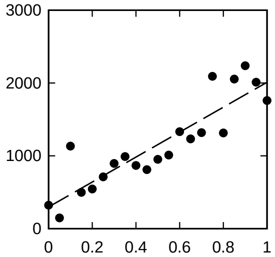
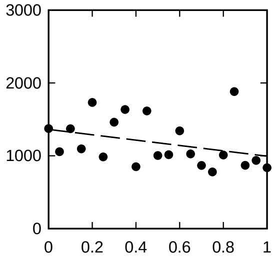
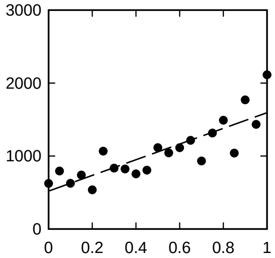
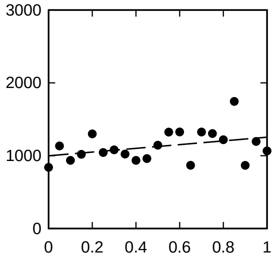
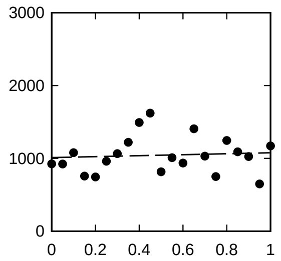
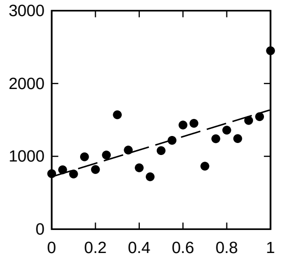
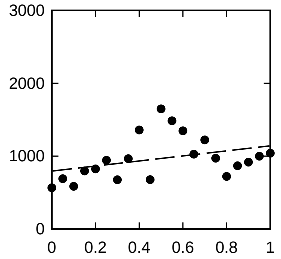
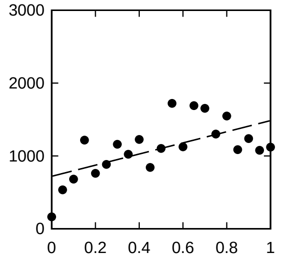
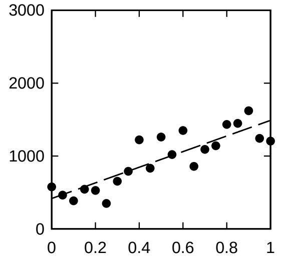
reach-shear-B	0.383	10%	51	255	

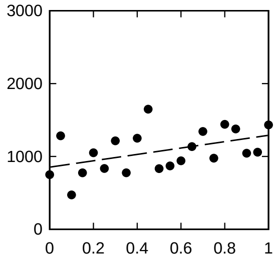
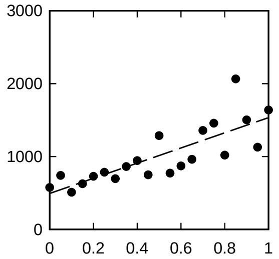
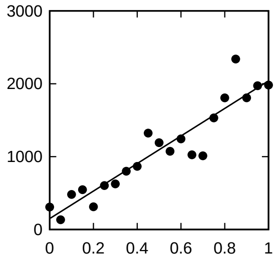
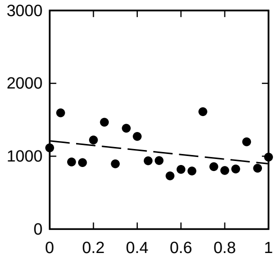
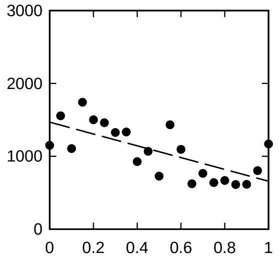
Table 33. Parameter sensitivity analysis results for trout parameters.

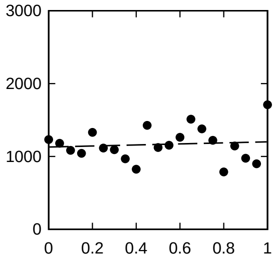
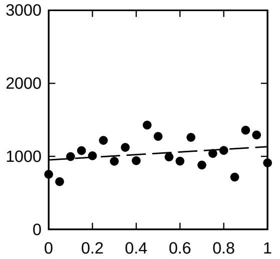
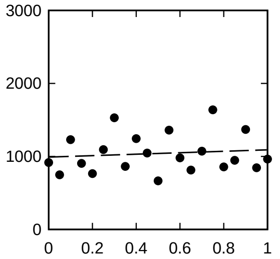
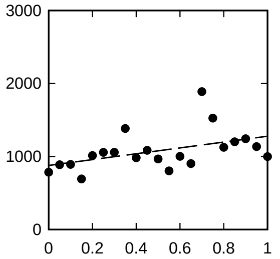
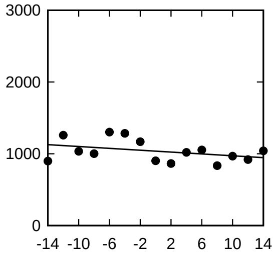
Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
trout-capture-R1	1.7	20%	900	2251	
trout-capture-R9	0.4	20%	365	911	
trout-cmax-A	0.628	50%	249	249	

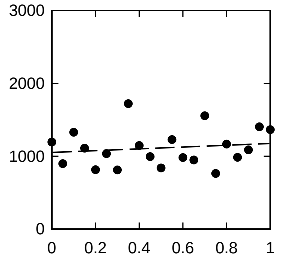
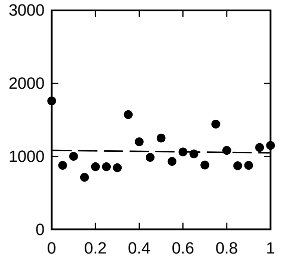
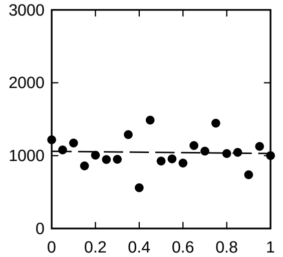
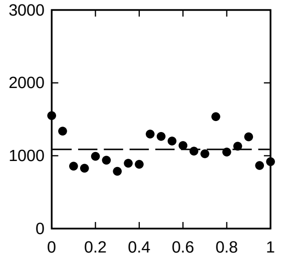
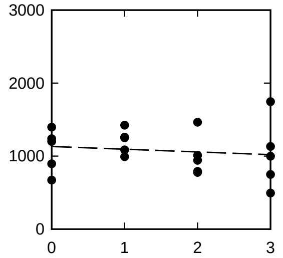
Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
trout-cmax-B	0.7	20%	1048	2619	
trout-emerge-length-mode, min, max (three parameters varied together)	2.8	40%	335	419	
trout-energy-density	5900	20%	-839	-2096	
trout-fitness-horizon	60	40%	372	465	
trout-fitness-length	***16.9***	***30%***	***85***	***142***	

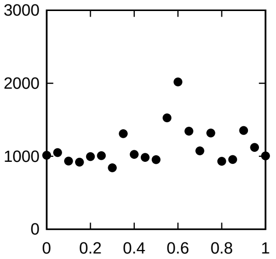
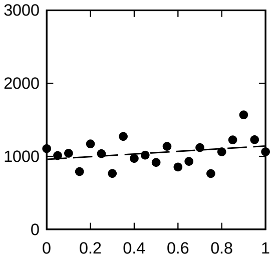
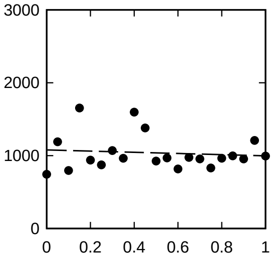
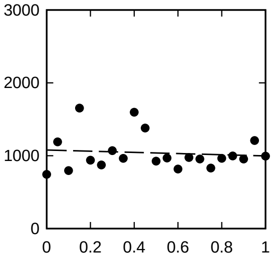
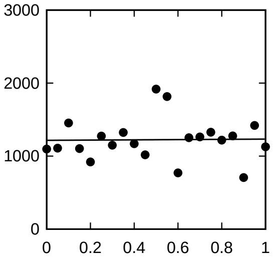
Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
trout-light-exp	-0.2	20%	177	442	
trout-light-min	0.5	40%	1712	2140	
trout-light-threshold	20	40%	-368	-460	
trout-max-speed-A	2.8	20%	1075	2687	
trout-max-speed-B	21	10%	255	1277	

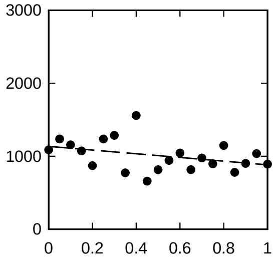
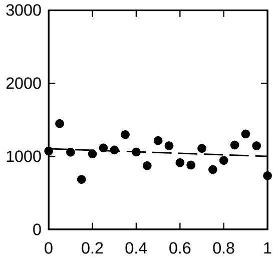
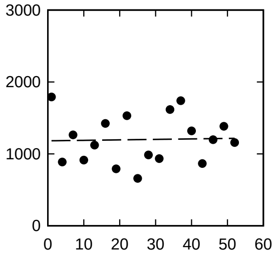
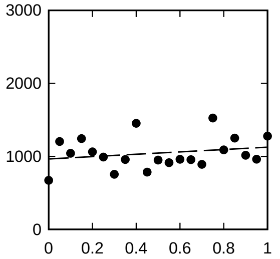
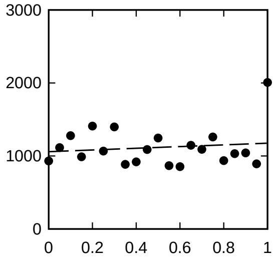
Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
trout-max-speed-C	-0.0029	10%	66	332	
trout-max-speed-D	0.084	10%	919	4596	
trout-max-speed-E	0.37	10%	1007	5037	
trout-move-dist-A	50	50%	1614*	1614	
trout-move-dist-B	2	40%	1073	1341	

Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
trout-pisciv-length	15	20%	437	1094	
trout-react-dist-A	4	40%	1041	1301	
trout-react-dist-B	2	20%	1890	4726	
trout-resp-A	36	20%	-314	-784	
trout-resp-B	0.783	10%	-810	-4726	

Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
trout-resp-C	0.002	10%	71	356	
trout-resp-D	1.4	10%	181	906	
trout-search-area	20000	40%	100	124	
trout-spawn-egg-viability	0.8	40%	400	500	
trout-spawn-end-day	5/31	Varied ±14 d	(Not comparable to other parameters)		

Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
trout-spawn-fecund-exp	2.51	10%	123	617	
trout-spawn-fecund-mult	0.18	20%	-34	-86	
trout-spawn-max-flow-change	0.2	20%	-30	-75	
trout-spawn-max-temp	13	20%	0	-1	
trout-spawn-min-age	1	Varied 0-3 y	(Not comparable to other parameters)		

Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
trout-spawn-min-length	13	30%	231	384	
trout-spawn-min-temp	7	20%	183	457	
trout-spawn-prob	0.04	50%	-81	-81	
trout-spawn-start-day	4/1	Varied ±14 d	(Not comparable to other parameters)		
trout-spawn-suitability-tol	0.1	80%	17	10	

Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
trout-spawn-wt-loss-fraction	0.2	40%	-253	-317	
trout-superind-max-length	5	40%	-105	-131	
trout-superind-max-rep	10	Varied 1-52 by 3	(Not comparable to other parameters)		
trout-turbid-exp	-0.116	20%	162	404	
trout-turbid-min	0.1	40%	117	146	

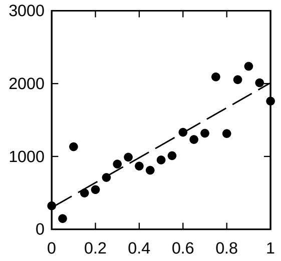
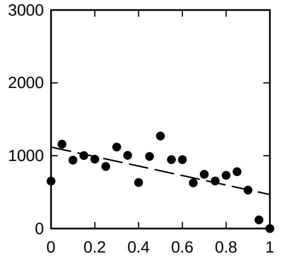
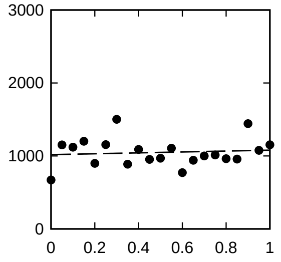
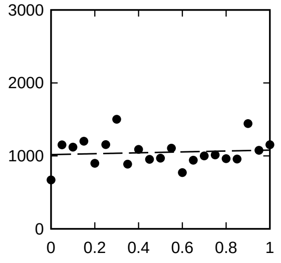
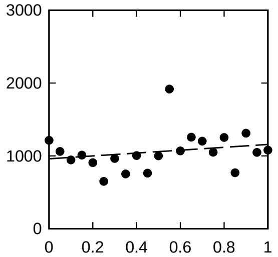
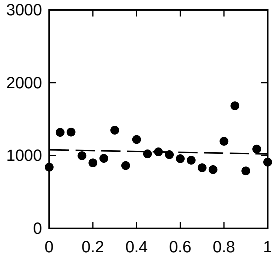
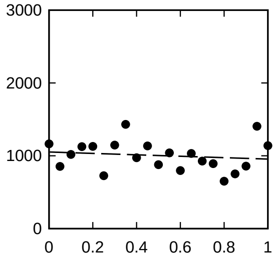
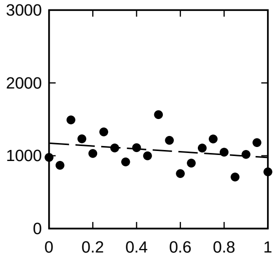
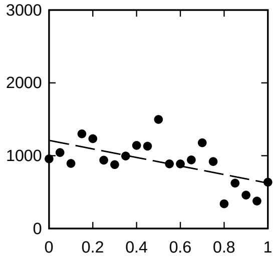
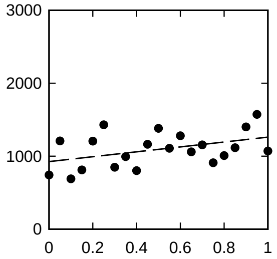
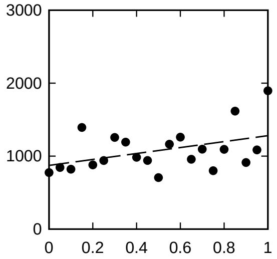
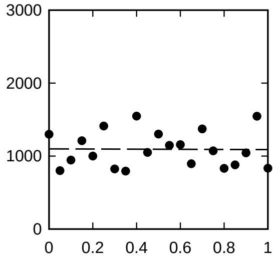
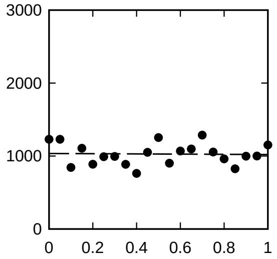
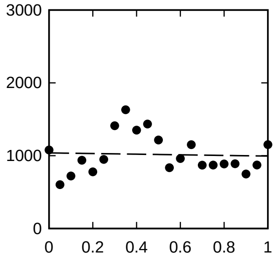
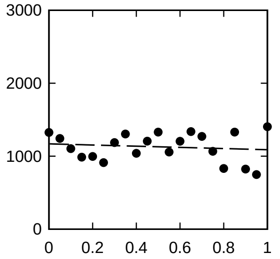
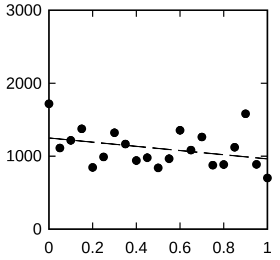
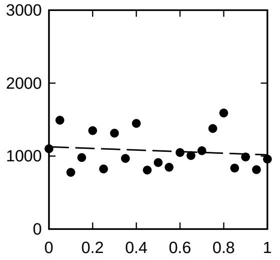
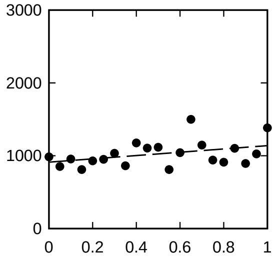
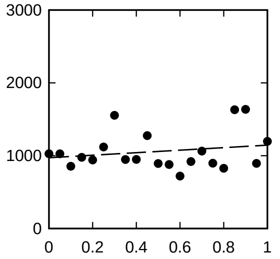
Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
trout-turbid-threshold	5	40%	1712	2140	

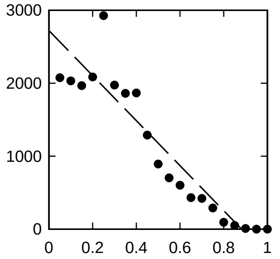
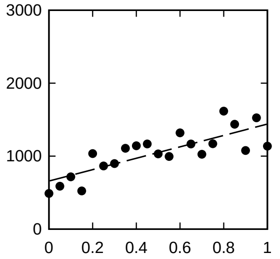
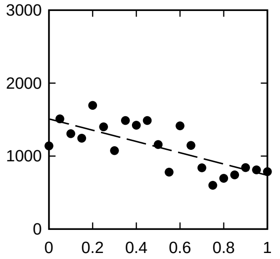
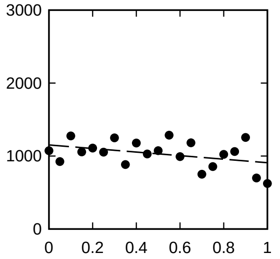
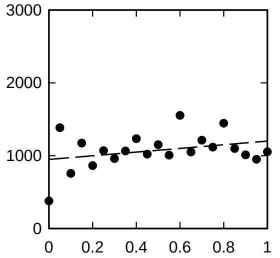
Table 34. Parameter sensitivity analysis results for trout survival parameters.

Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
mort-condition-S-at-K5	0.98	2%	-652	-16289	
mort-fish-pred-D1	35	30%	62	103	
mort-fish-pred-D9	5	30%	110	183	

Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
mort-fish-pred-hiding-factor	0.7	30%	197	329	
mort-fish-pred-I1	50	50%	-57	-57	
mort-fish-pred-I9	-50	50%	-98	-98	
mort-fish-pred-L1	3	40%	-197	-246	
mort-fish-pred-L9	6	40%	-1532 over the range 0.45-1.0	-1915	

Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
mort-fish-pred-P1	5.00E-06	50%	335	335	
mort-fish-pred-P9	5.00E-07	50%	408	408	
mort-fish-pred-T1	6	30%	-10	-16	
mort-fish-pred-T9	2	30%	-14	-24	
mort-high-temp-T1	30	10%	-43	-217	

Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
mort-high-temp-T9	25.8	10%	-81	-404	
mort-high-velocity-V1	1.8	20%	-287	-717	
mort-high-velocity-V9	1.4	20%	-110	-275	
mort-strand-survival-when-dry	0.5	20%	227	566	
mort-terr-pred-D1	0	30%	173	288	

Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
mort-terr-pred-D9	140	30%	-3085	-5142	
mort-terr-pred-H1	200	30%	781	1302	
mort-terr-pred-H9	-50	30%	-773	-1288	
mort-terr-pred-hiding-factor	0.8	20%	-244	-609	
mort-terr-pred-I1	50	50%	252	252	

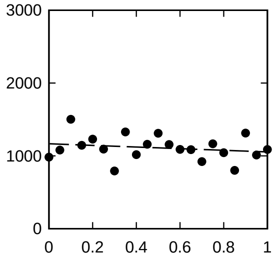
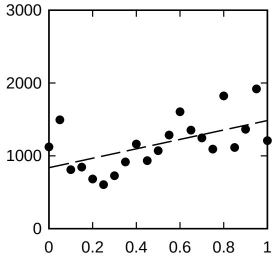
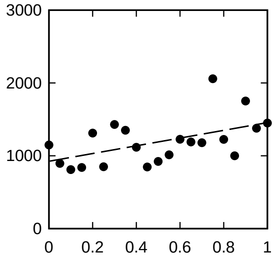
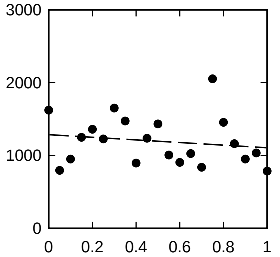
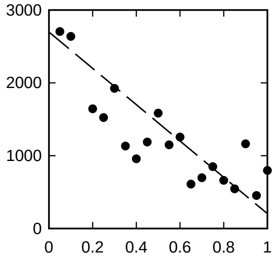
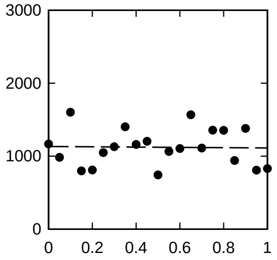
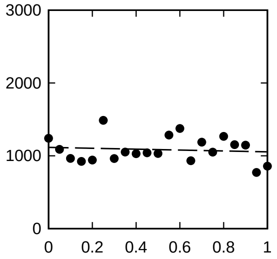
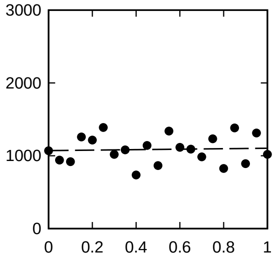
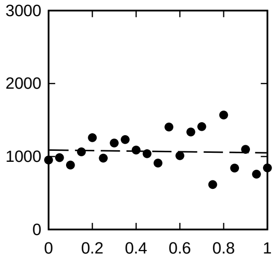
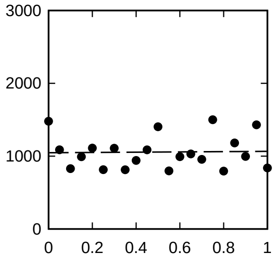
Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
mort-terr-pred-I9	-10	50%	-112	-112	
mort-terr-pred-L1	6	40%	647	809	
mort-terr-pred-L9	3	40%	531	664	
mort-terr-pred-V1	20	30%	-180	-300	
mort-terr-pred-V9	100	30%	-2494	-4157	

Table 35. Parameter sensitivity analysis results for redd parameters.

Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
redd-area	1200	40%	-20	-25	
redd-devel-A	-0.00025	1%	-61	-3075	
redd-devel-B	0.00134	1%	32	1607	
redd-devel-C	3.21E-05	0.1%	-38	-19196	
mort-redd-dewater-surv	0.9	10%	19	95	

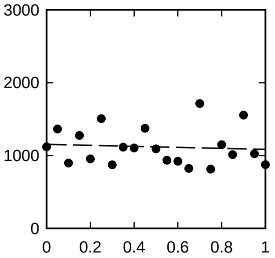
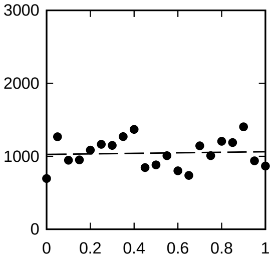
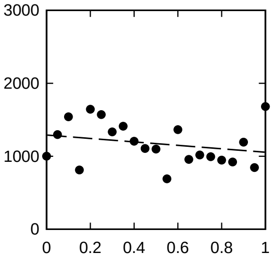
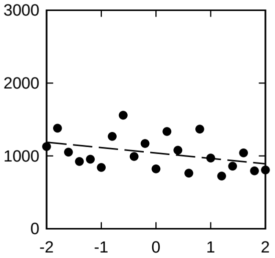
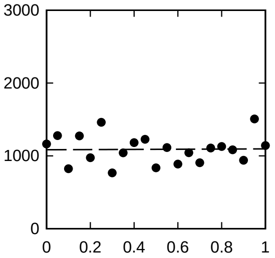
Parameter	Value	Range	Scaled slope	Unscaled slope	Graph
mort-redd-hi-temp-T1	30	10%	-70	-348	
mort-redd-hi-temp-T9	21	10%	37	184	
mort-redd-lo-temp-T1	-3	50%	-236	-236	
mort-redd-lo-temp-T9	0	Varied -2 to 2°C	(Not comparable to other parameters)		
mort-redd-scour-depth	5	50%	12	12	

Table 36. Parameters ranked by absolute value of scaled slope in the sensitivity analysis. Parameters near the top of this table can be assumed important in all applications; other parameters may have strong effects in particular applications.

Parameter	Scaled slope	Unscaled slope
reach-drift-conc	3705	3705
mort-terr-pred-D9	-3085	-5142
mort-terr-pred-V9	-2494	-4157
trout-react-dist-B	1890	4726
trout-light-min	1712	2140
trout-move-dist-A	1614	1614
mort-fish-pred-L9	-1532	-1915
reach-terr-pred-min	1435	7176
trout-max-speed-A	1075	2687
trout-move-dist-B	1073	1341
trout-cmax-B	1048	2619
trout-react-dist-A	1041	1301
trout-max-speed-E	1007	5037
trout-max-speed-D	919	4596
reach-drift-regen-distance	-916	-916
trout-capture-R1	900	2251
trout-energy-density	-839	-2096
trout-resp-B	-810	-4052
mort-terr-pred-H1	781	1302
mort-terr-pred-H9	-773	-1288
mort-condition-S-at-K5	-652	-16289

Parameter	Scaled slope	Unscaled slope
mort-terr-pred-L1	647	809
trout-spawn-min-length	630	1049
reach-fish-pred-min	553	2766
mort-terr-pred-L9	531	664
trout-pisciv-length	437	1094
mort-fish-pred-P9	408	408
trout-spawn-egg-viability	400	500
trout-fitness-horizon	372	465
trout-light-threshold	-368	-460
trout-capture-R9	365	911
trout-emerge-length-mode	335	419
mort-fish-pred-P1	335	335
trout-resp-A	-314	-784
max-spawn-flow	-304	-304
mort-redd-lo-temp-T9	-296	-74
mort-high-velocity-V1	-287	-717
reach-search-prod	-262	-262
trout-max-speed-B	255	1277
trout-spawn-wt-loss-fraction	-253	-317
mort-terr-pred-I1	252	252
trout-cmax-A	249	249
mort-terr-pred-hiding-factor	-244	-609
mort-redd-lo-temp-T1	-236	-236

Parameter	Scaled slope	Unscaled slope
reach-shading	232	580
mort-strand-survival-when-dry	227	566
mort-fish-pred-hiding-factor	197	329
mort-fish-pred-L1	-197	-246
reach-shear-A	-184	-461
trout-spawn-min-temp	183	457
trout-resp-D	181	906
mort-terr-pred-V1	-180	-300
trout-light-exp	177	442
mort-terr-pred-D1	173	288
trout-turbid-exp	162	404
latitude	-133	-333
light-correction	124	311
trout-spawn-fecund-exp	123	617
trout-turbid-min	117	146
mort-terr-pred-I9	-112	-112
mort-high-velocity-V9	-110	-275
mort-fish-pred-D9	110	183
trout-superind-max-length	-105	-131
trout-search-area	100	124
mort-fish-pred-I9	-98	-98
reach-light-turbid-coef	-83	-207
trout-spawn-prob	-81	-81

Parameter	Scaled slope	Unscaled slope
mort-high-temp-T9	-81	-404
trout-resp-C	71	356
light-at-night	-70	-44
mort-redd-hi-temp-T1	-70	-348
trout-max-speed-C	66	332
mort-fish-pred-D1	62	103
redd-devel-A	-61	-3075
mort-fish-pred-I1	-57	-57
reach-shear-B	51	255
reach-shelter-speed-frac	50	63
mort-high-temp-T1	-43	-217
trout-turbid-threshold	-40	50
redd-devel-C	-38	-19196
mort-redd-hi-temp-T9	37	184
trout-spawn-fecund-mult	-34	-86
redd-devel-B	32	1607
trout-spawn-max-flow-change	-30	-75
redd-area	-20	-25
mort-redd-dewater-surv	19	95
trout-spawn-suitability-tol	17	10
mort-fish-pred-T9	-14	-24
mort-redd-scour-depth	12	12
mort-fish-pred-T1	-10	-16

Parameter	Scaled slope	Unscaled slope
trout-spawn-max-temp	0	1

26.3 Sensitivity of primary predictions to initial conditions

Sensitivity of results to initial conditions is a common concern with simulation models. Modelers often talk about how long to let a model "warm up": how many of the first results of a simulation should be ignored because they remain strongly influenced by the initial state of the simulated population.

To address this question, we analyzed the sensitivity of inSTREAM 7's results to the initial population abundance. In seven scenarios, we varied the initial abundance of all ages of trout from 0.5 to 2.0 times the standard abundance, in steps of 0.25. We then simulated 10 years and examined how strongly the results at the end of each year were correlated to initial abundance. For replication, we repeated this experiment 5 times, each with a different shuffle of the water years (using the year shuffler described in Sect. 8.2). We conducted the experiment for two study sites: the small Little Jones Creek site used for parameter sensitivity experiments (Sect. 26.2), and the larger Clear Creek 3A site (the "RESTORED" site of Railsback et al. 2021). The Little Jones Creek site has a standard initial population of 212, 38, and 4 trout of ages 0, 1, and 2. The Clear Creek site's standard initial population is 1500, 350, and 100 trout of ages 0, 1, and 2. We analyzed the abundance of age 1 and older trout.

Results of this experiment varied strongly among replicates, but on average (Figure 69) they indicate that model results remain strongly affected by the assumed initial abundance through the first two simulated years, are only weakly affected by years 3-4, and are essentially independent of initial conditions by the fifth year.

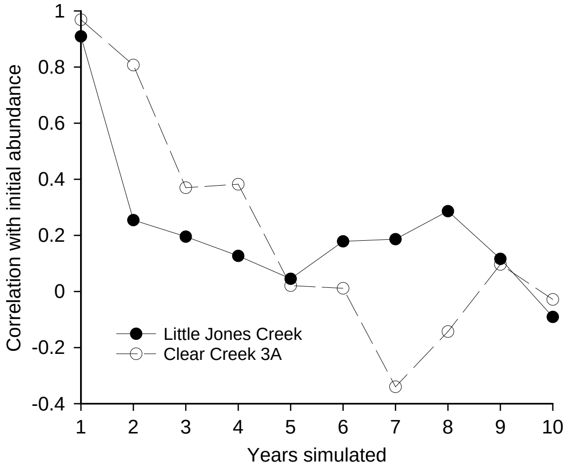


Figure 69. Results of the initial population sensitivity experiment. The Y axis indicates the mean correlation, over five year-shuffler replicates, of the abundance of age 1 and older trout in the simulated year versus the initial abundance.

Our general guidance is that inSTREAM results from the first two simulated years are likely to be strongly affected by initial conditions, while results after the fourth year are unlikely to be so affected. However, the persistence of initial condition effects can vary among model applications: sites where trout are assumed to spawn at older ages and survive longer are likely to have more “momentum” in population dynamics and take longer to be unaffected by initial conditions.

26.4 Sensitivity of primary predictions to spatial resolution

Our discussions of inSTREAM’s spatial resolution (Sect. 4.2.1) and habitat cell input (Sect. 22.1) identify several ways that the spatial resolution can affect model results. The effect of cell size on results was directly evaluated by Butcher and Parrish (2006; also described in Sect. 15.4.3 of Railsback et al. 2009). Butcher and Parrish used an earlier version (4.2) of inSTREAM that used rectangular habitat cells, and simply divided cells to make alternative habitat input ranging from few large cells to many small cells, with nothing else changed. In their study, making cells smaller than the initial resolution produced artifacts (presumably due to competition for the resources within a cell) that affected results, while making cells larger did not. This work confirms our guidance that cells should be no smaller than necessary to capture important spatial variation in habitat.

We cannot provide specific guidance on cell size, because cell size interacts with other variables—especially those controlling food availability and superindividual size—in limiting how many fish of what sizes a cell can support. Unfortunately, there is no simple way to address this question via simulation experiments with inSTREAM 7.

26.5 Robustness of management conclusions to parameter uncertainty

A key issue in any modeling study is understanding how robust results are to modeling assumptions and inputs: how different would the results, conclusions, and decisions from the model be if the model used a different equation, or was calibrated differently, or used different input? Traditionally these questions have been thought of in terms of model sensitivity and uncertainty, but it is more productive to think of them in terms of robustness: how robust are the conclusions drawn from a modeling study to its details (see Sect. 9.7 of Grimm and Railsback 2005; Grimm and Berger 2016)? This robustness question is especially important for complex models such as inSTREAM, but it cannot be answered without first answering several other questions:

- Robustness of what? Are we interested in the robustness of the primary predictions of inSTREAM—the simulated trout population abundance, production, etc.—or of secondary predictions such as the predicted differences among scenarios or the predicted sensitivity of population status to variables such as instream flow or temperature?
- Robustness to what? Are we interested in the robustness of results to equations and assumptions (e.g., which processes are included vs. ignored in inSTREAM), or to parameter values, or to input data?
- In what context? The robustness of results from inSTREAM undoubtedly varies with the conditions simulated. For example, results may be insensitive to equations and parameters for temperature mortality when inSTREAM is applied to a site where temperatures never exceed 15ºC, but very sensitive to these assumptions at sites with higher temperatures.

Evaluating the robustness of study results to the basic modeling assumptions and equations of inSTREAM will usually be beyond the scope and capability of routine applications. Making such an evaluation would require identifying reasonable alternative assumptions, implementing them in the software, testing them, and then analyzing the results of the alternative assumptions. That kind of analysis certainly represents interesting and valuable research but is impractical for most studies.

What kinds of robustness analyses are practical and valuable for routine applications of inSTREAM? For study designs involving comparison of several alternative scenarios or analyzing sensitivity of predicted trout populations to a small number of variables, it will often be practical to conduct additional model runs and examine how robust the most important results are to a few key inputs. Most important is determining whether and how the final ranking of management alternative scenarios changes as key inputs vary. (Drechsler et al. 2003 provide an example for a different kind of model.) These key inputs could include:

- Values for a small number of the parameters identified in Sect. 26.2 as having the strongest effects on the primary predictions of inSTREAM.
- Values of any additional parameters expected to be particularly important for the specific study, for example, parameters controlling sources of high mortality among fish and eggs.
- Hydrologic and weather conditions: years with high vs. low base flows, more vs. fewer extreme flow events, warm vs. cool temperatures, etc. Rates of mortality among simulated fish and eggs can again be used to identify important inputs to evaluate.
- The sequence of water years in the input (Sect. 8.2).

Analyzing robustness to key parameter values is especially easy and valuable for illustrating how inSTREAM’s mechanisms make conclusions based on its results insensitive to calibration details and parameter uncertainty. It has the additional value of illustrating when there is less difference among management alternatives than indicated by inSTREAM’s initial results. We recommend the following parameter robustness analysis steps for inSTREAM applications that involve ranking of management alternatives or other discrete scenarios.

1. Define the management alternative scenarios and identify 1-2 specific measures of predicted trout population benefits for ranking the scenarios. For example, the ranking measure could be the total biomass of adult trout on October 1 of simulation years 4-10; and, when relative abundance of multiple species is important, the fraction of adults of native vs. introduced species.
2. Identify a small number (typically, 2-5) of inSTREAM parameters to examine. Normally these will be parameters used for calibration (e.g., ***reach-drift-conc***, ***reach-terr-pred-min***; Sect. 24.3) because they are especially uncertain and important. But the robustness analysis could also use parameters especially relevant to a particular study. For example, Railsback et al. (2021) studied how the daily light cycle affected inSTREAM’s predicted population responses to flow, so they examined robustness to parameters that relate predation risk and food intake to light levels.
3. Select 2-3 values of each parameter, over wide but realistic ranges. The number of parameters and parameter values must be small enough to make it feasible to execute one model run for each parameter combination and each scenario. The number of model runs

is $S \times V^P$, where S is the number of scenarios, V is the number of values per parameter and P is the number of parameters. When only two values are used per parameter, they should be at the lower and upper ends of the range of realistic values; with three values, a central value can be included.

4. Execute model runs that use all combinations of all parameter values, for each scenario. The BehaviorSpace tool (Sect. 15) makes this easy.
5. Evaluate the ranking measures for each model run. For each parameter value combination, rank the scenarios by the values of the ranking measures (Excel's RANK function is useful for this).
6. Display the distribution of ranks by scenario. Examine the variation in rankings and how they overlap across scenarios. Management scenarios that overlap in ranking can be considered less robustly different in predicted trout population effects.

Railsback et al. (2021) illustrated this approach. They examined eight flow scenarios, referred to here simply as scenarios 1-8. They selected three values of two parameters, resulting in 72 model runs ($S \times V^P = 8 \times 3^2$). Then, for each of the nine sets of parameter values, they ranked the flow scenarios from 1 to 8, with 8 being the scenario with highest simulated trout biomass and 1 the least. The rankings (Figure 70) indicate the robustness of inSTREAM's results to uncertainty in the light parameters. Flow scenarios 1-2 consistently produced the lowest trout biomass, scenarios 3 and 8 were consistently the next two lowest, and scenario 6 almost always produced the most trout. The rank distributions of scenarios 4-7 overlapped broadly, indicating that the differences among those scenarios are not as robust, so the scenarios could be treated as having nearly equivalent benefits.

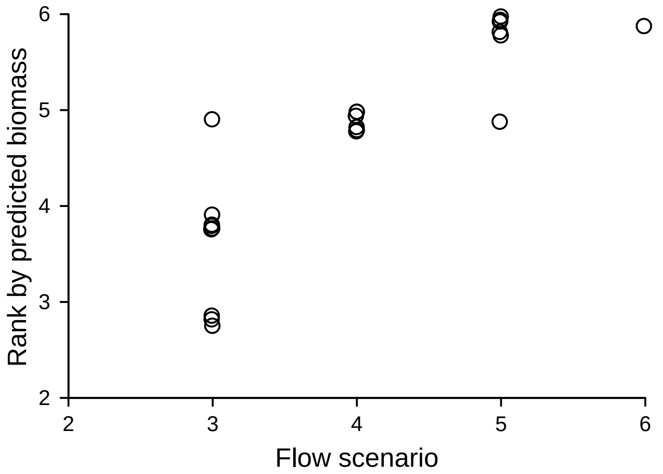


Figure 70. Results of an example analysis of scenario rankings to parameter uncertainty, re-drawn from Figure 5 of Railsback et al. (2021). Y values are "jittered" to make each symbol visible.

27 Validation of InSTREAM

27.1 What is "validation"?

Potential users of inSTREAM and its results, especially management decision-makers, are naturally concerned with validation: the extent to which inSTREAM's ability to predict population responses has been tested against data. This concern is natural because, despite its 20+ year history, inSTREAM can seem new and experimental compared to conventional models such as PHABSIM (Bovee et al. 1998). But questions about validation of inSTREAM also undoubtedly arise because it is clear that it *can* be tested in many ways; this is in sharp contrast to PHABSIM, which makes no directly testable predictions.

Even though inSTREAM can make many testable predictions—e.g., about population abundance, relative abundance of multiple species, age class structure, and even behavior and spatial distributions of individuals—validation of such complex models is not straightforward for several reasons. First, traditional validation of primary results such as population predictions is not convincing unless the model's internal mechanisms (e.g., habitat and activity selection in inSTREAM 7) have been tested successfully: a model could be calibrated to look successful at one site without having the right mechanisms to make it generally successful at many sites. Second, validation of models that make several kinds of specific predictions requires extensive and accurate data from the study site, for both model input and comparison to results. Collection of such data is expensive and not always feasible. Third, while inSTREAM is complex it is still a model, intentionally simplified in many ways; therefore, a real trout population can be affected by many processes and events that inSTREAM cannot predict. (We address these issues further below.)

These challenges to validation of complex IBMs are well-known, and the generally accepted way of addressing them is to use a "weight-of-evidence" approach to validation (Grimm and Railsback 2005; Augusiak et al. 2014; Grimm et al. 2020). Instead of treating comparison of primary predictions to data as a necessary and sufficient test of model validity, the weight-of-evidence approach looks at a wide range of evidence for model validity and usefulness. Chapter 10 of Railsback and Harvey (2020) provides general guidance on the weight-of-evidence approach. For inSTREAM, this wide range of evidence (summarized below) includes simulation experiments that tested key submodels against observed patterns and controlled field experiments that tested predictions at the individual and population levels.

Even with the extensive evaluation and validation done so far, validation of inSTREAM's population predictions remains a legitimate concern, and we hope that future applications include validation studies as often as possible. To support such applications, the following two subsections provide a summary of completed validation and evaluation studies and guidance for future studies.

27.2 Published validation studies

Here we summarize inSTREAM evaluation and validation studies published through 2020. We have not yet had the opportunity to test predictions of population response to a major instream flow change, but these studies provide evidence of inSTREAM's validity for that purpose.

Pattern-oriented validation of habitat and activity selection submodels. Throughout the evolution of inSTREAM, we have conducted simulation experiments that confirm its ability to produce realistic habitat and activity selection behaviors and their response to conditions such as food availability, predation risk, competition, and physical habitat. We focused on validating these behavior submodels because they represent the most important adaptive behavior and are the most innovative submodels. The validation studies (Railsback and Harvey 2002; Railsback et al. 2005, 2020) are discussed in sections 3.2.1 and 3.2.3. Each of these studies validated the submodels driving adaptive behavior by showing that they cause inSTREAM to reproduce observed patterns of response to a variety of factors.

Pattern-oriented analysis of population-level predictions. The simulation experiments of Railsback et al. (2002) showed that an early version of inSTREAM could reproduce a variety of general patterns observed in real trout populations. These experiments are summarized in Sect. 3.2.2.

Field evaluation of feeding and growth submodels. Harvey and Railsback (2014) validated inSTREAM's feeding and growth submodels by comparing predicted growth to observations of individual trout in a small, well-controlled field experiment. InSTREAM was calibrated to growth observed in a set of control habitat units and then used to predict growth in another set of habitat units with reduced flow. The growth predictions corresponded well with observed growth. This experiment also confirmed the importance of inSTREAM's search feeding mode as an alternative to drift feeding: growth predictions were more accurate when both search and drift feeding were simulated, compared to drift feeding only. This study produced a second piece of evidence for inSTREAM's validity: the drift food concentrations estimated by calibrating inSTREAM to observed growth matched concentrations measured at the study site.

Independent validation of terrestrial predation survival parameter values. InSTREAM's parameter controlling risk from terrestrial predators (***reach-terr-pred-min*** in inSTREAM 7; Sect. 9.18) has always been estimated only via calibration of the model to observed trout abundance, because predation risk is very difficult to observe and there has been no literature upon which to base parameter values. Typical values of the parameter for earlier versions of inSTREAM were in the range of 0.97 to 0.99. (Because it does not ignore risk during night, calibration of inSTREAM 7 typically produces lower values than these for ***reach-terr-pred-min***; Railsback et al. 2021.) To provide an independent estimate of this parameter, Harvey and Nakamoto (2013) conducted an extensive study of predation on trout confined to the high-risk conditions that ***reach-terr-pred-min*** represents. Their observations yielded survival estimates (0.98-0.99) within the narrow range of those obtained via calibration. This experiment provides evidence of the realism of inSTREAM's terrestrial predation submodel, which is among its most important (as indicated by the sensitivity analysis in Sect. 26.2).

Validation of predicted population effects of flow diversion. Harvey et al. (2014) calibrated two earlier versions of inSTREAM—representing daytime only, and day and night—to a small stream site upstream of a water diversion, and then used the calibrated parameter values to predict trout abundance and size at a second site downstream of the diversion. The models predicted several population characteristics well; these included the difference in biomass between sites, seasonal patterns in growth (low growth in the summer dry season), and patterns in growth among individuals. Both versions of inSTREAM indicated that the lower abundance downstream of the diversion was due in part to habitat differences and not only to reduced flow. (Subsequent to this study, a habitat improvement project was implemented downstream of the diversion; it led to much higher trout abundance.)

An unplanned validation: effects of barriers on population size structure. We used inSTREAM to evaluate effects of barrier density on trout population persistence in a small watershed (Harvey and Railsback 2012). Among the model results were populations isolated upstream of diversions having higher survival among juveniles but fewer and smaller adults, as a result of the habitat selection, growth, and survival submodels. In a field study of the same issue, Letcher et al. (2007) found the same pattern of age and size structure differences above and below barriers but attributed it possibly to an evolutionary response. Comparison of the simulation results of Harvey and Railsback (2012) to the field observations of Letcher et al. (2007) indicates that inSTREAM's submodels for growth, survival, and habitat selection together are capable of reproducing observed effects of barriers on key population characteristics, and that evolution is not always necessary to produce those effects.

27.3 Guidance for site-specific validation

We strongly encourage users of inSTREAM to conduct validation studies at their sites whenever feasible. In particular, applications to hydropower licensing that can support pre- and post-licensing monitoring are valuable opportunities to test the model analyses and predictions used to support licensing decisions. Doing so is in the spirit of adaptive management as a process that includes testing, evaluating, and improving the models used for management decisions (Walters 1986). Whether or not a validation study "proves" anything about inSTREAM's predictive ability, it almost certainly will produce useful knowledge about the model and the system it is applied to.

We base the following recommendations for validation studies on our experience with all versions of inSTREAM. Some of these recommendations also apply to field studies to collect calibration data.

Predict responses to major changes or differences between sites. InSTREAM is intended to predict trout population responses to change, so if nothing changes the model cannot be tested. Several of our hoped-for validation opportunities evaporated because nothing changed; one example was a hydropower licensing case in which the recommendation from inSTREAM (and other studies) was to not increase flows. Our successes with validation have come when we could calibrate inSTREAM to one site and then use it to predict conditions at adjacent sites that differ substantially in flow or physical habitat. Predicting population response to a future major change in flow at the same site should be equally productive. The key point is that validation success requires two sites, or two time periods at the same site, that (a) are similar enough so an inSTREAM calibration to one can be assumed valid at the other, yet (b) different enough (in flow regime, channel shape, etc.) to produce trout population differences large enough to detect both in the model and in the field. Simply monitoring one undisturbed site has not proven useful for validation because the year-to-year variation in abundance at the site is not large enough to distinguish from the combination of field data uncertainty and the uncertainty and stochasticity in model results.

Test lower-level predictions in addition to abundance. Traditionally, validation of population models is focused on abundance. However, inSTREAM makes many more kinds of predictions than abundance, and modern monitoring technologies such as PIT tags make it feasible to test predictions about characteristics such as the age and size structure of the population, growth and survival rates, and habitat selection. Attempting to validate lower-level predictions often helps understand why model predictions do or do not resemble observed population responses.

Be aware of data limitations and be sure that data are comparable to model results. Model validation studies tend to focus on uncertainties and errors in the model without adequate consideration of uncertainties and error in the field data. Data uncertainty and precision are one concern: we need to know the uncertainty or observation error in field measurements of trout abundance before we can test how well model predictions match those measurements. More precise field observations allow more rigorous model tests.

Comparability of field observations and model results are a second concern, especially with inSTREAM 7, because this model predicts how many trout are actively feeding versus hiding during each phase of the daily light cycle. Counts of trout feeding during daytime, e.g., from standard snorkel surveys, are therefore not comparable to inSTREAM abundance output. Ideally, observations of abundance will rely on methods that better count all trout. Censusing trout during both day and night is certainly better than daytime only, but even counting active trout throughout a 24-hour cycle has the problem that the same individuals may feed during several phases: it is not safe to assume that trout feeding during night are not also feeding during day or dusk.

We do not recommend attempting validation (or calibration) of inSTREAM 7 by comparing its predicted abundance of trout feeding in daytime to daytime field observations. The number of trout predicted to be feeding in any particular light phase is one of inSTREAM’s most uncertain results. Whether any simulated individual feeds or hides at any time is a complex prediction sensitive to a number of inputs and parameters; and multiple feeding patterns can provide almost equal fitness (Railsback et al. 2020). It does not appear that inSTREAM 7 needs to predict the percentage of fish feeding at different times of day accurately to make useful population predictions.

Comparability of uncertainty and variability measures is another issue that demands caution. Field census methods such as multiple-pass electrofishing and mark-recapture can produce confidence intervals on abundance estimates. InSTREAM also produces uncertainty measures, especially the variability among replicate simulations (Sect. 25.1). It can be informative for validation to see if model-predicted abundance is within the confidence limits on field-observed abundance, but the uncertainty of model results and that of field observations are not truly comparable. InSTREAM results have no measurement error (the output reports simulation results with 100% accuracy) so estimates of measurement uncertainty from field observations do not correspond with any simulation results. Conversely, replicates in inSTREAM represent only the effects of the model’s stochasticity, which has no analog in nature, so variability among replicates is not comparable to any field observations.

The overall recommendation is to be very careful in designing field studies and statistical analyses to ensure that any comparisons between field observations and model results use legitimately comparable variables or measures.

Limit expectations and expect differences that are not due to "error". Keep in mind that inSTREAM is a model, which means it is intentionally simplified and therefore cannot predict everything that can happen in nature. Validation studies must expect real trout populations to be affected by events and processes that are absent or highly simplified in inSTREAM, so the model will not always predict observations well. Examples of such events and processes that we suspect have affected our field populations include: episodic predation by otters, angler harvest, and acute influxes of fine sediment (e.g., following forest fires) that smothered redds and habitat. Such events can keep model predictions from matching data but do not make inSTREAM "wrong"—models cannot predict effects of events and processes they do not represent.

Ideally, validation studies are long enough to capture sufficient observations unaffected by unmodeled events. Study plans can incorporate the likelihood that not all years will produce data useful for testing the model. However, if events and processes not included in inSTREAM happen frequently enough to dominate population dynamics, then inSTREAM is not—unless modified—a useful model for the site (and neither is any other approach that assumes population dynamics are driven only by flow, temperature, turbidity, and physical habitat).

Plan to turn validation failures into opportunities for learning and improvement. A failure of inSTREAM to predict observed trout population dynamics can indicate that the model's representation of some process important at the site is too simple, wrong, or completely lacking. In that case, the ideal response is to learn what is inadequate or lacking, and improve inSTREAM by fixing it. In a simulation experiment not originally intended as a validation study, we found that inSTREAM predicted trout population extinction under elevated turbidity regimes that did not cause extinction of a real trout population (Harvey and Railsback 2009). This model "failure" indicated that inSTREAM lacked a mechanism that real trout use to feed in high turbidity. Laboratory experiments indicated that this mechanism was feeding close to the bottom and using non-visual senses to detect prey (Harvey and White 2008), which can easily be represented via inSTREAM's search feeding submodel.

This example illustrates how a particular process in inSTREAM can be especially important at particular study sites, so a simple submodel that suffices at other sites becomes inadequate. We encourage users to treat validation not as a test of whether inSTREAM is right or wrong, but instead as a way of determining what parts of it need improvement for their specific application.

Acknowledgments

Since 1999, development of inSTREAM and related models has been funded by the USDA Forest Service, the US Environmental Protection Agency, the US Bureau of Reclamation, the US Fish and Wildlife Service, the Electric Power Research Institute, Pacific Gas and Electric, Southern California Edison, and Argonne National Laboratory. Development of inSTREAM 7 was sponsored in part by the USDA Forest Service, Pacific Southwest Research Station and by Karlstad University, Sweden, with funding from Vattenfall AB.

The first version of inSTREAM was based in part on the trout IBM of Van Winkle et al. (1996, 1998), and inSTREAM 7 still contains some methods from that model.

For providing literature and data used in inSTREAM 7, with thank Prof. Erika Eliason, Department of Ecology, Evolution, & Marine Biology, University of California, Santa Barbara; and Dr. David Beauchamp, U.S. Geological Survey, Washington Cooperative Fish and Wildlife Research Unit.

We thank Eliza I. Gilbert (U.S. Fish and Wildlife Service), Mahboobeh Hajiesmaeili (Karlstad University, Sweden; Tarbiat Modares University, Iran), Peter Dudley (NOAA Southwest Fisheries Science Center, Fisheries Ecology Division) and Kwanmok Kim (NOAA Southwest Fisheries Science Center, Fisheries Ecology Division), for thoughtful and helpful reviews of this report.

References

Alexander, G.R. 1979. Predators of fish in coldwater streams. In: Clepper, H., ed. Predator-prey systems in fisheries management. Washington, D. C.: Sport Fishing Institute: 153-170.

Allouche, S. 2002. Nature and functions of cover for riverine fish. Bulletin Français de la Pêche et de la Pisciculture. 365/366: 297-324.

Alsop, D.H.; Wood, C.M. 1997. The interactive effects of feeding and exercise on oxygen consumption, swimming performance and protein usage in juvenile rainbow trout (*Oncorhynchus mykiss*). Journal of Experimental Biology. 200: 2337-2346.

Armstrong, J.D.; Griffiths, S.W. 2001. Density-dependent refuge use among over-wintering wild Atlantic salmon juveniles. Journal of Fish Biology. 58: 1524-1530.

Augusiak, J.; Van den Brink, P.J.; Grimm, V. 2014. Merging validation and evaluation of ecological models to 'evaludation': a review of terminology and a practical approach. Ecological Modelling. 280: 117-128.

Avery, E. L. 1985. Sexual maturity and fecundity of brown trout in central and northern Wisconsin streams. Technical Bulletin No. 154. On file with: Wisconsin Department of Natural Resources, 101 S Webster Street, PO Box 7921, Madison, WI 53707. 13 p.

Ayllón, D.; Nicola, G.G.; Elvira, B.; Almodóvar, A. 2019a. Optimal harvest regulations under conflicting tradeoffs between conservation and recreational fishery objectives. Fisheries Research. 216: 47-58. https://doi.org/10.1016/j.fishres.2019.03.021.

Ayllón, D.; Nicola, G.G.; Elvira, B.; Almodóvar, A. 2021. Climate change will render size-selective harvest of cold-water fish species unsustainable in Mediterranean freshwaters. Journal of Applied Ecology. 58: 562-575. https://doi.org/10.1111/1365-2664.13805.

Ayllón, D.; Railsback, S.F.; Almodóvar, A. [and others]. 2018. Eco-evolutionary responses to recreational fishing under different harvest regulations. Ecology and Evolution. 8(19): 9600-9613. https://doi.org/10.1002/ece3.4270.

Ayllón, D.; Railsback, S.F.; Harvey, B.C. [and others]. 2019b. Mechanistic simulations predict that thermal and hydrological effects of climate change on Mediterranean trout cannot be offset by adaptive behaviour, evolution, and increased food production. Science Of the Total Environment. 133648. https://doi.org/10.1016/j.scitotenv.2019.133648.

Ayllón, D.; Railsback, S.F.; Vincenzi, S. [and others]. 2016. InSTREAM-Gen: Modelling eco-evolutionary dynamics of trout populations under anthropogenic environmental change. Ecological Modelling. 326: 36-53. http://dx.doi.org/10.1016/j.ecolmodel.2015.07.026.

Behnke, R.J. 1992. Native trout of western North America. American Fisheries Society Monograph 6. Bethesda MD: American Fisheries Society. 275 p.

Booker, D.J.; Dunbar, M.J.; Ibbotson, A. 2004. Predicting juvenile salmonid drift-feeding habitat quality using a three-dimensional hydraulic-bioenergetic model. Ecological Modelling. 177: 157–177.

Bovee, K.D.; Lamb, B.L.; Bartholow, J.M. [and others]. 1998. Stream habitat analysis using the instream flow incremental methodology. Report USGS/BRD-1998-0004. Fort Collins, Colorado: U. S. Geological Survey, Biological Resources Division. 131 p.

Bozeman, B.B.; Grossman, G.D. 2019. Mechanics of foraging success and optimal microhabitat selection in Alaskan Arctic grayling (*Thymallus arcticus*). Canadian Journal of Fisheries and Aquatic Sciences. 76(5): 815-830.

Brauner, C.; Iwama, G.; Randall, D. 1994. The effect of short-duration seawater exposure on the swimming performance of wild and hatchery-reared juvenile coho salmon (*Oncorhynchus kisutch*) during smoltification. Canadian Journal of Fisheries and Aquatic Sciences. 51(10): 2188-2194.

Brauner, C.; Shrimpton, J.; Randall, D. 1992. Effect of short-duration seawater exposure on plasma ion concentrations and swimming performance in coho salmon (*Oncorhynchus kisutch*) parr. Canadian Journal of Fisheries and Aquatic Sciences. 49(11): 2399-2405.

Brown, H.W. 1974. Handbook of the effects of temperature on some North American fishes. Unpublished report. On file with: American Electric Power Service Corporation, 322 Dewalt Avenue SW, Canton, OH 44702. 524 p.

Brown, L.R.; Moyle, P.B. 1991. Changes in habitat and microhabitat partitioning within an assemblage of stream fishes in response to predation by Sacramento squawfish (*Ptychocheilus grandis*). Canadian Journal of Fisheries and Aquatic Sciences. 48: 849-856.

Butcher, G.; Parrish, M. 2006. Hydraulic simulations for regional fish modeling. Arcata, California: Humboldt State University. 143 p. M.S. thesis.

Butler, P.J.; Day, N.; Namba, K. 1992. Interactive effects of seasonal temperature and low pH on resting oxygen uptake and swimming performance of adult brown trout *Salmo trutta*. Journal of Experimental Biology. 165: 195-212.

Carlander, K.D. 1969. Handbook of freshwater fishery biology. 3rd ed. Ames, Iowa: The Iowa State University Press. 752 p.

Carlson, S.M.; Seamons, T.R. 2008. A review of quantitative genetic components of fitness in salmonids: implications for adaptation to future change. Evolutionary Applications. 1(2): 222-238. https://doi.org/10.1111/j.1752-4571.2008.00025.x.

Cienciala, P.; Hassan, M.A. 2013. Linking spatial patterns of bed surface texture, bed mobility, and channel hydraulics in a mountain stream to potential spawning substrate for small resident trout. Geomorphology. 197: 96-107.

Collis, K.; Roby, D.D.; Craig, D.P. [and others]. 2001. Colonial waterbird predation on juvenile salmonids tagged with passive integrated transponders in the Columbia River estuary: vulnerability of different salmonid species, stocks, and rearing types. Transactions of the American Fisheries Society. 130(3): 385-396.

Cunjak, R.A.; Prowse, T.D.; Parrish, D.L. 1998. Atlantic salmon (*Salmo salar*) in winter: "the season of parr discontent"? Canadian Journal of Fisheries and Aquatic Sciences. 55(Suppl. 1): 161-180.

Cunningham, P.M. 2007. A sensitivity analysis of an individual-based trout model. Arcata, CA: Humboldt State University. 89 p. M.S. thesis.

DeRobertis, A.; Ryer, C.H.; Veloza, A.; Brodeur, R.D. 2003. Differential effects of turbidity on prey consumption of piscivorous and planktivorous fish. Canadian Journal of Fisheries and Aquatic Sciences. 60: 1517-1526.

Deslauriers, D.; Chipps, S.R.; Breck, J.E. [and others]. 2017. Fish Bioenergetics 4.0: An R-Based Modeling Application. Fisheries. 42(11): 586-596. https://doi.org/10.1080/03632415.2017.1377558.

DeVries, P. 1997. Riverine salmonid egg burial depths: review of published data and implications for scour studies. Canadian Journal of Fisheries and Aquatic Sciences. 54: 1685-1698.

Diana, J.S.; Hudson, J.P.; Richard D. Clark, J. 2004. Movement patterns of large brown trout in the mainstream Au Sable River, Michigan. Transactions of the American Fisheries Society. 133: 34-44.

Dickerson, B.R.; Vinyard, G.L. 1999. Effects of high chronic temperatures and diel temperature cycles on the survival and growth of Lahontan cutthroat trout. Transactions of the American Fisheries Society. 128: 516-521.

Donofrio, E.; Simon, T.; Neuswanger, J.R.; Grossman, G.D. 2018. Velocity and dominance affect prey capture and microhabitat selection in juvenile Chinook (*Oncorhynchus tshawytscha*). Environmental Biology of Fishes. 101(4): 609-622. https://doi.org/10.1007/s10641-018-0723-5.

Drechsler, M.; Frank, K.; Hanski, I. [and others]. 2003. Ranking metapopulation extinction risk: from patterns in data to conservation management decisions. Ecological Applications. 13(4): 990-998.

Dudley, P.N. 2018. A salmonid individual-based model as a proposed decision support tool for management of a large regulated river. Ecosphere. 9(1): e02074. https://doi.org/10.1002/ecs2.2074.

Duffie, J.A.; Beckman, W.A. 2013. Solar engineering of thermal processes. 4th ed. Hoboken, New Jersey: John Wiley & Sons. 910 p.

Eliason, E.J.; Clark, T.D.; Hinch, S.G.; Farrell, A.P. 2013a. Cardiorespiratory collapse at high temperature in swimming adult sockeye salmon. Conservation Physiology. 1: 10.1093. https://doi.org/10.1093/conphys/cot008.

Eliason, E.J.; Clark, T.D.; Hinch, S.G.; Farrell, A.P. 2013b. Cardiorespiratory performance and blood chemistry during swimming and recovery in three populations of elite swimmers: Adult sockeye salmon. Comparative Biochemistry and Physiology, Part A. 166: 385-397. https://doi.org/10.1016/j.cbpa.2013.07.020.

Elliott, J.M. 1982. The effects of temperature and ration size on growth and energetics of salmonids in captivity. Comparative Biochemistry and Physiology. 73: 81-91.

Elliott, J.M. 1994. Quantitative ecology and the brown trout. New York: Oxford University Press. 286 p.

Embody, G.C. 1934. Relation of temperature to the incubation periods of eggs of four species of trout. Transactions of the American Fisheries Society. 64(1): 281-292.

Essington, T.E.; Quinn, T.P.; Ewert, V.E. 2000. Intra- and inter-specific competition and the reproductive success of sympatric Pacific salmon. Canadian Journal of Fisheries and Aquatic Sciences. 57: 205-213.

Essington, T.E.; Sorensen, P.W.; Paron, D.G. 1998. High rate of redd superimposition by brook trout (*Salvelinus fontinalis*) and brown trout (*Salmo trutta*) in a Minnesota stream cannot be explained by habitat availability alone. Canadian Journal of Fisheries and Aquatic Sciences. 55: 2310-2316.

Farrell, A. 2007. Cardiorespiratory performance during prolonged swimming tests with salmonids: a perspective on temperature effects and potential analytical pitfalls. Philosophical Transactions of the Royal Society B: Biological Sciences. 362(1487): 2017-2030.

Farrell, A.; Lee, C.; Tierney, K. [and others]. 2003. Field-based measurements of oxygen uptake and swimming performance with adult Pacific salmon using a mobile respirometer swim tunnel. Journal of Fish Biology. 62(1): 64-84.

Fausch, K.D. 1984. Profitable stream positions for salmonids: relating specific growth rate to net energy gain. Canadian Journal of Zoology. 62: 441-451.

Finstad, A.G.; Einum, S.; Forseth, T.; Ugedal, O. 2007. Shelter availability affects behaviour, size-dependent and mean growth of juvenile Atlantic salmon. Freshwater Biology. 52: 1710-1718. https://doi.org/10.1111/j.1365-2427.2007.01799.x.

Fleming, I. A.; Reynolds, J. D. 2004. Salmonid breeding systems. In: Hendry, A. P.; and Stearns, S. C., eds. Evolution illuminated, salmon and their relatives. Oxford: Oxford University Press: 264-294.

Forbes, V.E.; Railsback, S.; Accolla, C. [and others]. 2019. Predicting impacts of chemicals from organisms to ecosystem service delivery: A case study of endocrine disruptor effects on trout. Science of the Total Environment. 649: 949-959. https://doi.org/10.1016/j.scitotenv.2018.08.344.

From, J.; Rasmussen, G. 1984. A growth model, gastric evacuation, and body composition in rainbow trout, *Salmo gairdneri* Richardson, 1836. Dana. 3: 61-139.

Gallaugher, P.; Thorarensen, H.; Kiessling, A.; Farrell, A. 2001. Effects of high intensity exercise training on cardiovascular function, oxygen uptake, internal oxygen transport and osmotic balance in chinook salmon (Oncorhynchus tshawytscha) during critical speed swimming. Journal of Experimental Biology. 204(16): 2861-2872.

Gard, M. 1997. Techniques for adjusting spawning depth habitat utilization curves for availability. Rivers. 6(2): 94-102.

Garshelis, D.L. 2000. Delusions in habitat evaluation: measuring use, selection, and importance. In: Boitani, L.; Fuller, T.K., eds. Researchg Techniques in Animal Ecology, Controversies and Consequences. Methods and Cases in Conservation Science. New York: Columbia University Press: 111-164.

Gauthey, Z.; Lang, M.; Elosegi, A. [and others]. 2017. Brown trout spawning habitat selection and its effects on egg survival. Ecology of Freshwater Fish. 26(1): 133-140.

Glarner, H. 2011. Length of day and twilight. www.gandraxa.com/length_of_day.xml. (13 October 2021).

Gortázar, J.; Alonso, C.; García de Jalón, D. 2012. Brown trout redd superimposition in relation to spawning habitat availability. Ecology of Freshwater Fish. 21(2): 283-292.

Gregory, R.S.; Levings, C.D. 1998. Turbidity reduces predation on migrating juvenile pacific salmon. Transactions of the American Fisheries Society. 127(2): 275-285. https://doi.org/10.1577/1548-8659(1998)127<0275:TRPOMJ>2.0.CO;2

Grémillet, D.; Nazirides, T.; Nikolaou, H.; Crivelli, A.J. 2012. Fish are not safe from great cormorants in turbid water. Aquatic Biology. 15(2): 187-194.

Griffiths, J.S.; Alderdice, D.F. 1972. Effects of acclimation and acute temperature experience on the swimming speed of juvenile coho salmon. Journal of the Fisheries Research Board of Canada. 29: 251-264.

Grimm, V.; Berger, U. 2016. Robustness analysis: Deconstructing computational models for ecological theory and applications. Ecological Modelling. 326: 162-167. http://dx.doi.org/10.1016/j.ecolmodel.2015.07.018.

Grimm, V.; Berger, U.; Bastiansen, F. [and others]. 2006. A standard protocol for describing individual-based and agent-based models. Ecological Modelling. 198: 115-296. https://doi.org/10.1016/j.ecolmodel.2006.04.023rai.

Grimm, V.; Berger, U.; DeAngelis, D.L. [and others]. 2010. The ODD protocol: a review and first update. Ecological Modelling. 221: 2760-2768.

Grimm, V.; Johnston, A.S.A.; Thulke, H.H. [and others]. 2020. Three questions to ask before using model outputs for decision support. Nature Communications. 11(1): 4959 https://doi.org/10.1038/s41467-020-17785-2.

Grimm, V.; Railsback, S.F. 2005. Individual-based modeling and ecology. Princeton Series in Theoretical and Computational Biology. Princeton, New Jersey: Princeton University Press. 428 p.

Grimm, V.; Railsback, S.F.; Vincenot, C.E. [and others]. 2020. The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. Journal of Artificial Societies and Social Simulation. 23(2): 7. https://doi.org/10.18564/jasss.4259.

Hajiesmaeili, M. 2019. Modification of the inSTREAM model using bioenergetics approach in order to simulate habitat suitability in rivers (Case study: Brown trout in Elarm River, Lar National Park). Tehran, Iran: Tarbiat Modares University. 278 p. Ph.D. dissertation.

Hansen, A.G.; Beauchamp, D.A. 2015. Latitudinal and photic effects on diel foraging and predation risk in freshwater pelagic ecosystems. Journal of Animal Ecology. 84: 532-544. https://doi.org/10.1111/1365-2656.12295.

Hansen, A.G.; Beauchamp, D.A.; Schoen, E.R. 2013. Visual prey detection responses of piscivorous trout and salmon: effects of light, turbidity, and prey size. Transactions of the American Fisheries Society. 142(3): 854-867.

Hanson, P.; Johnson, T.; Kitchell, J.; Schindler, D.E. 1997. Fish Bioenergetics 3.0. Madison, Wisconsin: University of Wisconsin Sea Grant Institute.

Harvey, B.C.; Nakamoto, R.J. 2013. Seasonal and among-stream variation in predator encounter rates for fish prey. Transactions of the American Fisheries Society. 142(3): 621-627. https://doi.org/10.1080/00028487.2012.760485.

Harvey, B.C.; Nakamoto, R.J.; White, J.L. 1999. Influence of large woody debris and a bankfull flood on movement of adult resident coastal cutthroat trout (*Oncorhynchus clarki*) during fall and winter. Canadian Journal of Fisheries and Aquatic Sciences. 56(11): 2161-2166.

Harvey, B.C.; Nakamoto, R.J.; White, J.L.; Railsback, S.F. 2014. Effects of streamflow diversion on a fish population: combining empirical data and individual-based models in a site-specific evaluation. North American Journal of Fisheries Management. 34: 247–257. https://doi.org/10.1080/02755947.2013.860062.

Harvey, B.C.; Railsback, S.F. 2007. Estimating multi-factor cumulative watershed effects on fish populations with an individual-based model. Fisheries. 32(6): 292-298.

Harvey, B.C.; Railsback, S.F. 2009. Exploring the persistence of stream-dwelling trout populations under alternative real-world turbidity regimes with an individual-based model. Transactions of the American Fisheries Society. 138: 348-360.

Harvey, B.C.; Railsback, S.F. 2012. Effects of passage barriers on demographics and stability properties of a virtual trout population. River Research and Applications. 28: 479-489. https://doi.org/10.1002/rra.1574.

Harvey, B.C.; Railsback, S.F. 2014. Feeding modes in stream salmonid population models: is drift feeding the whole story? Environmental Biology of Fishes. 97(5): 615-625. https://doi.org/10.1007/s10641-013-0186-7.

Harvey, B.C.; Railsback, S.F. 2021. “All fish, all the time”: a good general objective for fish passage projects? Fisheries. 46: 119-124.

Harvey, B.C.; Stewart, A.J. 1991. Fish size and habitat depth relationships in headwater streams. Oecologia. 87: 336-342.

Harvey, B.C.; White, J.L. 2008. Use of benthic prey by salmonids under turbid conditions in a laboratory stream. Transactions of the American Fisheries Society. 137: 1756-1763.

Harvey, B.C.; White, J.L. 2016. Use of cover for concealment behavior by rainbow trout: influences of cover structure and area. North American Journal of Fisheries Management. 36(6): 1308-1314. https://doi.org/10.1080/02755947.2016.1207728.

Harvey, B.C.; White, J.L. 2017. Axes of fear for stream fish: water depth and distance to cover. Environmental Biology of Fishes. 100(5): 565–573. https://doi.org/10.1007/s10641-017-0585-2.

Harwood, A.J.; Metcalfe, N.B.; Griffiths, S.W.; Armstrong, J.D. 2002. Intra- and inter-specific competition for winter concealment habitat in juvenile salmonids. Canadian Journal of Fisheries and Aquatic Sciences. 59(9): 1515-1523. https://doi.org/10.1139/f02-119.

Haschenburger, J.K. 1999. A probability model of scour and fill depths in gravel-bed channels. Water Resources Research. 35(9): 2857-2869.

Hawkins, D.K.; Quinn, T.P. 1996. Critical swimming velocity and associated morphology of juvenile coastal cutthroat trout (*Oncorhynchus clarki clarki*), steelhead trout (*Oncorhynchus mykiss*), and their hybrids. Canadian Journal of Fisheries and Aquatic Sciences. 53: 1487-1496.

Hayes, J.W.; Stark, J.D.; Shearer, K.A. 2000. Development and test of a whole-lifetime foraging and bioenergetics growth model for drift-feeding brown trout. Transactions of the American Fisheries Society. 129: 315-332.

Healey, M.C. 1991. Life history of chinook salmon (*Oncorhynchus tshawytscha*). In: Groot, C.; Margolis, L., eds. Pacific Salmon Life Histories. Vancouver: UBC Press: 564 p.

Hendry, A.P.; Morbey, Y.E.; Berg, O.K.; Wenburg, J.K. 2003. Adaptive variation in senescence: reproductive lifespan in a wild salmon population. Proceedings of the Royal Society of London B. 271: 259-266.

Hilborn, R. 1997. Statistical hypothesis testing and decision theory in fisheries science. Fisheries. 22(10): 19-20.

Hilborn, R.; Mangel, M. 1997. The ecological detective: confronting models with data. Monographs in Population Biology. Princeton NJ: Princeton University Press. 315 p.

Hilderbrand, R.H.; Kershner, J.L. 2000. Movement patterns of stream-resident cutthroat trout in Beaver Creek, Idaho–Utah. Transactions of the American Fisheries Society. 129(5): 1160-1170.

Hill, J.; Grossman, G.D. 1993. An energetic model of microhabitat use for rainbow trout and rosyside dace. Ecology. 74(3): 685-698.

Hodgens, L.S.; Blumenshine, S.C.; Bednarz, J.C. 2004. Great blue heron predation on stocked rainbow trout in an Arkansas tailwater fishery. North American Journal of Fisheries Management. 24(1): 63-75.

Hokanson, K.E.F.; Kleiner, C.F.; Thorslund, T.W. 1977. Effects of constant temperatures and diel temperature fluctuations on specific growth and mortality rates and yield of juvenile rainbow trout, (*Salmo gairdneri*). Journal of the Fisheries Research Board of Canada. 34: 639-648.

Hostetter, N.J.; Evans, A.F.; Roby, D.D.; Collis, K. 2012. Susceptibility of juvenile steelhead to avian predation: the influence of individual fish characteristics and river conditions. Transactions of the American Fisheries Society. 141(6): 1586-1599.

Houston, A.I.; McNamara, J.M. 1999. Models of adaptive behaviour: an approach based on state. Cambridge: Cambridge University Press. 378 p.

Hughes, N.F. 1992. Selection of positions by drift-feeding salmonids in dominance hierarchies: Model and test for arctic grayling (*Thymallus arcticus*) in subarctic mountain streams, interior Alaska. Canadian Journal of Fisheries and Aquatic Sciences. 49: 1999-2008.

Hughes, N.F.; Hayes, J.W.; Shearer, K.A.; Young, R.B. 2003. Testing a model of drift-feeding using 3-d videography of wild brown trout in a New Zealand river. Canadian Journal of Fisheries and Aquatic Sciences. 60: 1462-1476.

Imre, I.; Grant, J.W.; Cunjak, R. 2010. Density-dependent growth of young-of-the-year Atlantic salmon (*Salmo salar*) revisited. Ecology of Freshwater Fish. 19(1): 1-6.

Jager, H.I.; DeAngelis, D.L.; Sale, M.J. [and others]. 1993. An individual-based model for smallmouth bass reproduction and young-of-year dynamics in streams. Rivers. 4(2): 91-113.

Johansen, J.L.; Akanyeti, O.; Liao, J.C. 2020. Oxygen consumption of drift-feeding rainbow trout: the energetic tradeoff between locomotion and feeding in flow. The Journal of Experimental Biology. 223(12): jeb220962. https://doi.org/10.1242/jeb.220962.

Johnson, B.M.; Pate, W.M.; Hansen, A.G. 2017. Energy density and dry matter content in fish: new observations and an evaluation of some empirical models. Transactions of the American Fisheries Society. 146(6): 1262-1278. https://doi.org/10.1080/00028487.2017.1360392.

Johnson, M.D. 2007. Measuring habitat quality: a review. The Condor. 109(3): 489-504. https://doi.org/10.1650/8347.1.

Jones, M.W.; Hutchings, J.A. 2002. Individual variation in Atlantic salmon fertilization success: implications for effective population size. Ecological Applications. 12: 184-193.

Jonsson, B.; Jonsson, N. 2011. Ecology of Atlantic salmon and brown trout: habitat as a template for life histories. Dordrecht: Springer. 708 p.

Julian, J.P.; Doyle, M.W.; Stanley, E.H. 2008. Empirical modeling of light availability in rivers. Journal of Geophysical Research. 113: G03022. https://doi.org/10.1029/2007JG000601.

June, J.A. 1981. Life history and habitat utilization of cutthroat trout (*Salmo clarki*) in a headwater stream on the Olympic Peninsula, Washington. Seattle: University of Washington. 115 p. M.S. thesis.

Källo, K.; Birnie-Gauvin, K.; Jepsen, N.; and Aarestrup, K. 2023. Great cormorant (*Phalacrocorax carbo sinensis*) predation on adult anadromous brown trout (*Salmo trutta*). Ecology of Freshwater Fish. 32: 488-495.

Keeley, E.R.; Grant, J.W.A. 2001. Prey size of salmonid fishes in streams, lakes, and oceans. Canadian Journal of Fisheries and Aquatic Sciences. 58: 1122-1132.

Kepler, M.V.; Wagner, T.; Sweka, J.A. 2014. Comparative bioenergetics modeling of two Lake Trout morphotypes. Transactions of the American Fisheries Society. 143(6): 1592-1604.

Knapp, R.A.; Preisler, H.K. 1999. Is it possible to predict habitat use by spawning salmonids? A test using California golden trout (*Oncorhynchus mykiss* aguabonita). Canadian Journal of Fisheries and Aquatic Sciences. 56(9): 1576-1584.

Kondolf, G.M. 2000. Assessing salmonid spawning gravel quality. Transactions of the American Fisheries Society. 129: 262-281.

Lam, T. J. 1988. Environmental influences on gonadal activity in fish. In: Hoar, W.S.; Randall, D.J.; Donaldson, E.M., eds. Fish physiology, Volume IX, Reproduction, Part B: Behavior and fertility control. New York: Academic Press: 65-116.

Lee, C.G.; Farrell, A.P.; Lotto, A. [and others]. 2003. The effect of temperature on swimming performance and oxygen consumption in adult sockeye (Oncorhynchus nerka) and coho (O. kisutch) salmon stocks. Journal of Experimental Biology. 206(18): 3239-3251.

Letcher, B.H.; Nislow, K.H.; Coombes, J.A. [and others]. 2007. Population response to habitat fragmentation in a stream-dwelling brook trout population. PLoS ONE. 2(11): e1139.

Lien, L. 1978. The energy budget of the brown trout population of Øvre Heimdalsvatn. Ecography. 1(2-3): 279-300.

Lloyd, D.S.; Koenings, J.P.; Laperriere, J.D. 1987. Effects of turbidity in fresh waters of Alaska. North American Journal of Fisheries Management. 7(1): 18-33.

Louhi, P.; Mäki-Petäys, A.; Erkinaro, J. 2008. Spawning habitat of Atlantic salmon and brown trout: general criteria and intragravel factors. River Research and Applications. 24(3): 330-339. https://doi.org/10.1002/rra.1072.

MacNutt, M.J.; Hinch, S.G.; Farrell, A.P.; Topp, S. 2004. The effect of temperature and acclimation period on repeat swimming performance in cutthroat trout. Journal of Fish Biology. 65: 342-353.

MacNutt, M.J.; Hinch, S.G.; Lee, C.G. [and others]. 2006. Temperature effects on swimming performance, energetics, and aerobic capacities of mature adult pink salmon (Oncorhynchus gorbuscha) compared with those of sockeye salmon (Oncorhynchus nerka). Canadian Journal of Zoology. 84(1): 88-97.

Magee, J.P.; McMahon, T.E.; Thurow, R.F. 1996. Spatial variation in spawning habitat of cutthroat trout in a sediment-rich stream basin. Transactions of the American Fisheries Society. 125(5): 768-779.

Magnusson, W.E. 2000. Error bars: are they the king's clothes? Bulletin of the Ecological Socity of America. 81(2): 147-150.

Mangel, M.; Clark, C.W. 1986. Toward a unified foraging theory. Ecology. 67(5): 1127-1138.

Martin, B.T.; Pike, A.; John, S.N. [and others]. 2017. Phenomenological vs. biophysical models of thermal stress in aquatic eggs. Ecology Letters. 20(1): 50-59. https://doi.org/10.1111/ele.12705.

May, C.L.; Pryor, B.; Lisle, T.E.; Lang, M. 2009. Coupling hydrodynamic modeling and empirical measures of bed mobility to predict the risk of scour and fill of salmon redds in a large regulated river. Water Resources Research. 45(5): W05402. https://doi.org/10.1029/2007WR006498.

McNeil, W.J. 1964. Redd superimposition and egg capacity of pink salmon spawning beds. Journal of the Fisheries Board of Canada. 21(6): 1385-1396.

Mesa, M.G.; Weiland, L.K.; Christiansen, H.E. [and others]. 2013. Development and evaluation of a bioenergetics model for Bull Trout. Transactions of the American Fisheries Society. 142(1): 41-49.

Metcalfe, N.B.; Fraser, N.H.C.; Burns, M.D. 1999. Food availability and the nocturnal vs. diurnal foraging trade-off in juvenile salmon. Journal of Animal Ecology. 68: 371-381.

Meyer, K.A.; Schill, D.J.; Elle, F.S.; Lamansky, J.A.J. 2003. Reproductive demographics and factors that influence length at sexual maturity of Yellowstone cutthroat trout in Idaho. Transactions of the American Fisheries Society. 132: 183-195.

Michel, C.J.; Henderson, M.J.; Loomis, C.M. [and others]. 2020. Fish predation on a landscape scale. Ecosphere. 11(6): e03168.

Morán, P.; Labbé, L.; Garcia de Leaniz, C. 2016. The male handicap: male-biased mortality explains skewed sex ratios in brown trout embryos. Biology Letters. 12(12): 20160693.

Morbey, Y.E.; Ydenberg, R.C. 2003. Timing games in the reproductive phenology of female Pacific salmon (*Oncorhynchus* spp.). American Naturalist. 161(2): 284-298.

Moyle, P.B.; Baltz, D.M. 1985. Microhabitat use by an assemblage of California stream fishes: Developing criteria for instream flow determinations. Transactions of the American Fisheries Society. 114: 695-704.

Myrick, C.A. 1998. Temperature, genetic, and ration effects on juvenile rainbow trout (*Oncorhynchus mykiss*) bioenergetics. Davis, CA: University of California, Davis. 165 p. PhD.

Myrick, C.A.; Cech, J.J., Jr. 2000. Temperature influences on California rainbow trout physiological performance. Fish Physiology and Biochemistry. 22: 245-254.

Myrick, C.A.; Cech, J.J., Jr. 2003. The physiological performance of golden trout at water temperatures of 10 – 19 °C. California Fish and Game. 89(1): 20-29.

Myrick, C.A.; Cech, J.J., Jr. 2004. Temperature effects on juvenile anadromous salmonids in California's central valley: what don't we know? Reviews in Fish Biology and Fisheries. 14: 113-123.

Nakano, S.; Fausch, K.D.; Kitano, S. 1999. Flexible niche partitioning via a foraging mode shift: a proposed mechanism for coexistence in stream-dwelling charrs. Journal of Animal Ecology. 68(6): 1079-1092. https://doi.org/10.1046/j.1365-2656.1999.00355.x.

Naman, S.M.; Rosenfeld, J.S.; Jordison, E. [and others]. 2020. Exploitation of velocity gradients by sympatric stream salmonids: basic insights and implications for instream flow. North American Journal of Fisheries Management. 40(2): 320-329. https://doi.org/10.1002/nafm.10411.

Nelson, R.L.; Platts, W.S.; Casey, O. 1987. Evidence for variability in spawning behavior of interior cutthroat trout in response to environmental uncertainty. The Great Basin Naturalist. 47(3): 480-487.

Nislow, K.H.; Folt, C.L.; Parrish, D.L. 1999. Favorable foraging locations for young Atlantic salmon: application to habitat and population restoration. Ecological Applications. 9(3): 1085-1099.

O'Brien, W.J.; Barfield, M.; Sigler, K. 2004. The functional response of drift-feeding Arctic grayling: the effects of prey density, water velocity, and location efficiency. Canadian Journal of Fisheries and Aquatic Sciences. 58: 1957-1963.

Ojanguren, A. F.; Braña, F. 2003. Thermal dependence of embryonic growth and development in brown trout. Journal of Fish Biology 62: 580-590.

Ovidio, M.; Baras, E.; Goffaux, D. [and others]. 2002. Seasonal variations of activity pattern of brown trout (*Salmo trutta*) in a small stream, as determined by radio-telemetry. Hydrobiologia. 470(1): 195-202.

Pacific Gas and Electric Company (PG&E). 1994. Evaluation of factors causing variability in habitat suitability criteria for Sierra Nevada trout. Unpublished report 009.4-94.5. On file with: Pacific Gas and Electric Company, Department of Research and Development, 3400 Crow Canyon Road, San Ramon, CA 94583. 35 p.

Parra, I.; Nicola, G.G.; Vøllestad, L.A. [and others]. 2014. Latitude and altitude differentially shape life history trajectories between the sexes in non-anadromous brown trout. Evolutionary Ecology. 28(4): 707-720. https://doi.org/10.1007/s10682-014-9702-2.

Piccolo, J.J.; Hughes, N.F.; Bryant, M.D. 2008. Water velocity influences prey detection and capture by drift-feeding juvenile coho salmon (*Oncorhynchus kisutch*) and steelhead (*Oncorhynchus mykiss irideus*). Canadian Journal of Fisheries and Aquatic Sciences. 65: 266-275.

Power, M.E. 1987. Predator avoidance by grazing fishes in temperate and tropical streams: Importance of stream depth and prey size. In: Kerfoot, W.C.; Sih, A., eds. Predation: Direct and Indirect Impacts on Aquatic Communities. Hanover: University Press of New England: 333-352.

Quinn, T.P.; Buck, G.B. 2001. Size- and sex-selective mortality of adult sockeye salmon: bears, gulls, and fish out of water. Transactions of the American Fisheries Society. 130: 995–1005.

Railsback, S.F. 2000. Instream flow assessment methods: guidance for evaluating instream flow needs in hydropower licensing. Technical Report 1000554. On file with: Electric Power Research Institute, 3412 Hillview Avenue, Palo Alto, CA 94304. 226 p.

Railsback, S.F. 2016. Why it is time to put PHABSIM out to pasture. Fisheries. 41(12): 720-725. https://doi.org/10.1080/03632415.2016.1245991.

Railsback, S.; Ayllón, D.; Berger, U. [and others]. 2017. Improving execution speed of models implemented in NetLogo. Journal of Artificial Societies and Social Simulation. 20(1): 3. DOI: 10.18564/jasss.3282.

Railsback, S.F.; Gard, M.; Harvey, B.C. [and others]. 2013. Contrast of degraded and restored stream habitat using an individual-based salmon model. North American Journal of Fisheries Management. 33(2): 384-399. DOI: 10.1080/02755947.2013.765527.

Railsback, S.F.; Grimm, V. 2019. Agent-based and individual-based modeling: a practical introduction, 2nd edition. Princeton, New Jersey: Princeton University Press. 329 p.

Railsback, S.F.; Harvey, B.C. 2001. Individual-based model formulation for cutthroat trout, Little Jones Creek, California. Albany, CA: Pacific Southwest Research Station, Forest Service, U. S. Department of Agriculture. 80 p.

Railsback, S.F.; Harvey, B.C. 2002. Analysis of habitat selection rules using an individual-based model. Ecology. 83(7): 1817-1830.

Railsback, S.F.; Harvey, B.C. 2011. Importance of fish behaviour in modelling conservation problems: food limitation as an example. Journal of Fish Biology. 79: 1648–1662.

Railsback, S.F.; Harvey, B.C. 2013. Trait-mediated trophic interactions: is foraging theory keeping up? Trends in Ecology & Evolution. 28(2): 119-125. DOI: 10.1016/j.tree.2012.08.023.

Railsback, S.F.; Harvey, B.C. 2020. Modeling populations of adaptive individuals. Monographs in Population Biology 63. Princeton, New Jersey: Princeton University Press. 329 p.

Railsback, S.F.; Harvey, B.C.; Ayllón, D. 2020. Contingent tradeoff decisions with feedbacks in cyclical environments: testing alternative theories. Behavioral Ecology. 31(5): 1192-1206. DOI: 10.1093/beheco/araa070.

Railsback, S.F.; Harvey, B.C.; Ayllón, D. 2021. Importance of the daily light cycle in population-habitat relations: a simulation study. Transactions of the American Fisheries Society. 150(1): 130-143. DOI: 10.1002/tafs.10283.

Railsback, S.F.; Harvey, B.C.; Hayse, J.W.; LaGory, K.E. 2005. Tests of theory for diel variation in salmonid feeding activity and habitat use. Ecology. 86(4): 947-959.

Railsback, S.F.; Harvey, B.C.; Jackson, S.K.; Lamberson, R.H. 2009. InSTREAM: the individual-based stream trout research and environmental assessment model. Albany, California: USDA Forest Service, Pacific Southwest Research Station. 254 p.

Railsback, S.F.; Harvey, B.C.; Lamberson, R.H. [and others]. 2002. Population-level analysis and validation of an individual-based cutthroat trout model. Natural Resource Modeling. 15(1): 83-110.

Railsback, S.F.; Harvey, B.C.; White, J.L. 2014. Facultative anadromy in salmonids: linking habitat, individual life history decisions, and population-level consequences. Canadian Journal of Fisheries and Aquatic Sciences. 71(8): 1270-1278. DOI: 10.1139/cjfas-2014-0091.

Railsback, S.F.; Harvey, B.C.; White, J.L. 2015. Effects of spatial extent on modeled relations between habitat and salmon spawning success. Transactions of the American Fisheries Society. 144(6): 1220-1236. https://doi.org/10.1080/00028487.2015.1079553.

Railsback, S.F.; Hayse, J.W.; LaGory, K.E. 2006. Simulation analysis of within-day flow fluctuation effects on trout below Flaming Gorge Dam. Technical Report ANL/EVS/TM/06-01. Argonne, IL: Department of Energy, Argonne National Laboratory. 146 p. https://doi.org/10.2172/925305.

Railsback, S.F.; Kadvany, J. 2004. Demonstration flow assessment: procedures for judgement-based instream flow studies. Technical Report 1005389. On file with: Electric Power Research Institute, 3412 Hillview Avenue, Palo Alto, CA 94304. 124 p.

Railsback, S.F.; Kadvany, J. 2008. Demonstration flow assessment: judgment and visual observation in instream flow studies. Fisheries. 33(5): 217-227.

Railsback, S.F.; Lamberson, R.H.; Harvey, B.C.; Duffy, W.E. 1999. Movement rules for spatially explicit individual-based models of stream fish. Ecological Modelling. 123(2-3): 73-89.

Railsback, S.F.; Rose, K.A. 1999. Bioenergetics modeling of stream trout growth: temperature and food consumption effects. Transactions of the American Fisheries Society. 128: 241-256.

Railsback, S. F.; Sheppard, C. 2014. inSTREAM-SD: The Individual-based Stream Trout Research and Environmental Assessment Model with Sub-Daily Time Step, Version 6.1, model description. Unpublished report. On file with: Lang, Railsback & Associates, 250 California Avenue, Arcata, CA 95521. 131 p.

Railsback, S.F.; Stauffer, H.B.; Harvey, B.C. 2003. What can habitat preference models tell us? Tests using a virtual trout population. Ecological Applications. 13(6): 1580-1594.

Rand, P.S.; Stewart, D.J.; Seelbach, P.W. [and others]. 1993. Modeling steelhead population energetics in lakes Michigan and Ontario. Transactions of the American Fisheries Society. 122: 977-1001.

Rao, G.M.M. 1968. Oxygen consumption of rainbow trout (*Salmo gairdneri*) in relation to activity and salinity. Canadian Journal of Zoology. 46: 781-786.

Réalis-Doyelle, E.; Pasquet, A.; De Charleroy, D. [and others]. 2016. Strong effects of temperature on the early life stages of a cold stenothermal fish species, brown trout (*Salmo trutta* L.). PLoS ONE. 11(5): e0155487.

Reiser, D.W.; White, R.G. 1983. Effects of complete redd dewatering on salmonid egg-hatching success and development of juveniles. Transactions of the American Fisheries Society. 112: 532-540.

Rosenberger, A.E.; Dunham, J.B.; Neuswanger, J.R.; Railsback, S.F. 2015. Legacy effects of wildfire on stream thermal regimes and rainbow trout ecology: an integrated analysis of observation and individual-based models. Freshwater Science. 34(4): 1571–1584. https://doi.org/10.1086/683338.

Rottiers, D.V.; Tucker, R.M. 1982. Proximate composition and caloric content of eight Lake Michigan fishes. Washington, D.C.: U. S. Department of the Interior, Fish and Wildlife Service. 8 p.

Salecker, J.; Sciaini, M.; Meyer, K.M.; Wiegand, K. 2019. The NLRX R package: A next-generation framework for reproducible NetLogo model analyses. Methods in Ecology and Evolution. 10(11): 1854-1863. https://doi.org/10.1111/2041-210X.13286.

Schmidt, D.; O'Brien, W.J. 1982. Planktivorous feeding ecology of Arctic grayling (*Thymallus arcticus*). Canadian Journal of Fisheries and Aquatic Sciences. 39: 475-482.

Schneider, M.J.; Connors, T.J. 1982. Effects of elevated water temperature on the critical swimming speeds of yearling rainbow trout, *Salmo gairdneri*. Journal of Thermal Biology. 7: 227-229.

Simpkins, D.G.; Hubert, W.A.; Martinez Del Rio, C.; Rule, D.C. 2003a. Interacting effects of water temperature and swimming activity on body composition and mortality of fasted juvenile rainbow trout. Canadian Journal of Zoology. 81: 1641–1649.

Simpkins, D.G.; Hubert, W.A.; Martinez Del Rio, C.; Rule, D.C. 2003b. Physiological responses of juvenile rainbow trout to fasting and swimming activity: effect of body composition and condition indices. Transactions of the American Fisheries Society. 132: 576–589.

Sloat, M.R.; Reeves, G.H. 2014. Individual condition, standard metabolic rate, and rearing temperature influence steelhead and rainbow trout (*Oncorhynchus mykiss*) life histories. Canadian Journal of Fisheries and Aquatic Sciences. 71: 491-501. https://doi.org/10.1139/cjfas-2013-0366.

Smith, E.P.; Rose, K.A. 1995. Model goodness-of-fit analysis using regression and related techniques. Ecological Modelling. 77: 49-64.

Strange, E.M.; Moyle, P.B.; Foin, T.C. 1992. Interactions between stochastic and deterministic processes in stream fish community assembly. Environmental Biology of Fishes. 36: 1-15.

Sweka, J.A.; Hartman, K.J. 2001. Influence of turbidity on brook trout reactive distance and foraging success. Transactions of the American Fisheries Society. 130: 138-146.

Taylor, S.E.; Egginton, S.; Taylor, E.W. 1996. Seasonal temperature acclimatisation of rainbow trout: cardiovascular and morphometric influences on maximal sustainable exercise level. Journal of Experimental Biology. 199: 835-845.

Theurer, F.D.; Voos, K.A.; Miller, W.J. 1984. Instream water temperature model. Instream Flow Information Paper 16. FWS/OBS-84/15. Washington, D.C.: U. S. Fish and Wildlife Service. 321 p.

Thut, R.N. 1970. Feeding habits of the dipper in southwestern Washington. The Condor. 72(2): 234-235.

Tudorache, C.; Viaene, P.; Blust, R. [and others]. 2008. A comparison of swimming capacity and energy use in seven European freshwater fish species. Ecology of Freshwater Fish. 17(2): 284-291. https://doi.org/10.1111/j.1600-0633.2007.00280.x.

Utne-Palm, A.C. 2002. Visual feeding of fish in a turbid environment: physical and behavioural aspects. Marine and Freshwater Behaviour and Physiology. 35(1-2): 111-128.

Valdimarsson, S.K.; Metcalfe, N.B.; Thorpe, J.E.; Huntingford, F.A. 1997. Seasonal changes in sheltering: Effect of light and temperature on diel activity in juvenile salmon. Animal Behavior. 54: 1405-1412.

Van Winkle, W.; Jager, H.I.; Holcomb, B.D. 1996. An individual-based instream flow model for coexisting populations of brown and rainbow trout. Technical Report 106258. On file with: Electric Power Research Institute, 3412 Hillview Avenue, Palo Alto, CA 94304. 176 p.

Van Winkle, W.; Jager, H.I.; Railsback, S.F. [and others]. 1998. Individual-based model of sympatric populations of brown and rainbow trout for instream flow assessment: model description and calibration. Ecological Modelling. 110: 175-207.

Verhille, C.E.; English, K.K.; Cocherell, D.E. [and others]. 2016. High thermal tolerance of a rainbow trout population near its southern range limit suggests local thermal adjustment. Conservation Physiology. 4(1): cow057. https://doi.org/10.1093/conphys/cow057.

Vinyard, G.L.; O'Brien, W.J. 1976. Effects of light and turbidity on the reactive distance of bluegill (*Lepomis macrochirus*). Journal of the Fisheries Research Board of Canada. 33: 2845-2849.

Vogel, J.L.; Beauchamp, D.A. 1999. Effects of light, prey size, and turbidity on reaction distances of lake trout (*Salvelinus namaycush*) to salmonid prey. Canadian Journal of Fisheries and Aquatic Sciences. 56(7): 1293-1297.

Walters, C.J. 1986. Adaptive management of renewable resources. New York: Macmillan. 374 p.

White, J.W.; Rassweiler, A.; Samhouri, J.F. [and others]. 2014. Ecologists should not use statistical significance tests to interpret simulation model results. Oikos. 123(4): 385-388. https://doi.org/10.1111/j.1600-0706.2013.01073.x.

Whiterod, N.S.; Meredith, S.N.; Humphries, P. 2013. Refining the activity component of a juvenile fish bioenergetics model to account for swimming costs. Marine and Freshwater Behaviour and Physiology. 46(4): 201-210. https://doi.org/10.1080/10236244.2013.819155.

Wilcock, P.R.; Barta, A.F.; Shea, C.C. [and others]. 1996. Observations of flow and sediment entrainment on a large gravel-bed river. Water Resources Research. 32(9): 2897-2909.

Wilensky, U. 1999. NetLogo. http://ccl.northwestern.edu/netlogo/. Evanston, IL: Center for Connected Learning and Computer-based Modeling, Northwestern University.

Wilensky, U.; Rand, W. 2015. An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo. Cambridge, Massachusetts: MIT Press. 482 p.

Wood, J.L.; Fraser, D.J. 2015. Similar plastic responses to elevated temperature among different-sized brook trout populations. Ecology. 96(4): 1010-1019.

Xu, C.; Letcher, B.H.; Nislow, K.H. 2010. Context-specific influence of water temperature on brook trout growth rates in the field. Freshwater Biology. 55(11): 2253-2264. https://doi.org/10.1111/j.1365-2427.2010.02430.x.

Young, K.A. 2003. Evolution of fighting behavior under asymmetric competition: an experimental test with juvenile salmonids. Behavioral Ecology. 14: 127-134.

Zimmer, M.P.; Power, M. 2006. Brown trout spawning habitat selection preferences and redd characteristics in the Credit River, Ontario. Journal of Fish Biology. 68(5): 1333-1346. https://doi.org/10.1111/j.0022-1112.2006.00995.x.

Index