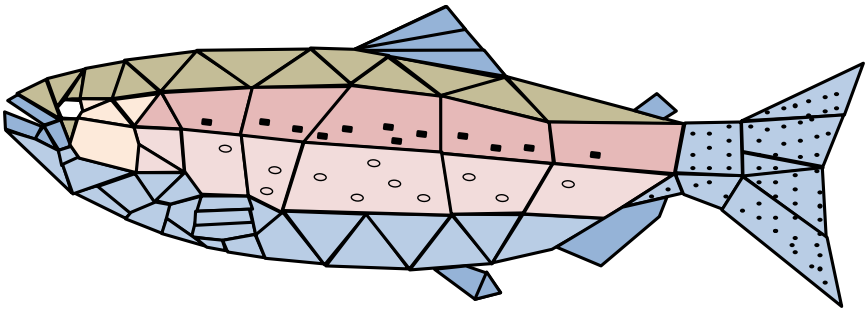
inSALMO Version 1.0: Model Improvements and Demonstration Application to Chinook Salmon Spawning, Incubation, and Rearing in Clear Creek, California

Final report

Contract R09PS20027
Improvement of Salmon Life-Cycle Framework Model (inSALMO)

Conducted by
Lang, Railsback & Associates, Arcata CA

For
U. S. Bureau of Reclamation, Mid-Pacific Regional Office, Sacramento CA



Prepared by:

Steven F. Railsback
Lang, Railsback & Associates

Bret C. Harvey, Jason L. White
USDA Forest Service
Pacific Southwest Research Station

Final report dated: October 14, 2011

Table of Contents

List of Figures iv
List of Tables vii
1 Introduction 1
1.1 Document objectives 1
1.2 *inSALMO* background 1
1.3 Project objectives 2
1.4 Document overview 3
1.5 Acknowledgements and credits 4
2 Model Description 5
3 Software Installation and Use 6
3.1 Installing *inSALMO* 6
3.2 Using the Graphical User Interface 6
3.3 Using *inSALMO* without the GUI 6
3.4 Modifying and compiling the source code 7
4 Example Application to Clear Creek 9
4.1 Study site selection 9
4.2 Habitat input and parameters 9
4.2.1 Time series inputs: flow, temperature, and turbidity 9
4.2.2 Cell geometry 10
4.2.3 Hydraulics 10
4.2.4 Shear stress parameters 11
4.3 Salmon input and parameter values 13
4.3.1 Spawner initialization data 13
4.3.2 Fall Chinook parameter values 14
4.4 Sensitivity analysis 18
4.4.1 Methods 18
4.4.2 Results 20
4.4.3 Sensitivity analysis conclusions 29
4.5 Calibration and validation 30
4.5.1 Redd locations 30
4.5.2 Outmigration timing 32
4.5.3 Outmigrant sizes 33
4.5.4 Total outmigrant numbers and contributions by size-class 34
5 Example Decision-support Applications 36

5.1 Evaluation of habitat restoration 36
5.1.1 Purpose and methods 36
5.1.2 Results 37
5.1.3 Conclusions 41
5.2 Evaluation of alternative instream flow and temperature regimes: Before 1995 vs. after 2001 42
5.2.1 Purpose and methods 42
5.2.2 Results 43
5.2.3 Conclusions 45
5.3 Comparison of OCAP climate change scenarios 45
5.3.1 Purpose and methods 45
5.3.2 Results 46
5.3.3 Conclusions 48
6 Limiting Factors Analysis for the Clear Creek Application 49
6.1 Objectives 49
6.2 Methods: Application of the Limiting Factors Tool 49
6.3 Results 49
6.4 Conclusions concerning limiting factors 56
7 Development of *inSALMO* Input from Instream Flow Study Data 57
7.1 Background and objectives 57
7.2 Data interpretation methods 57
7.2.1 Extraction of information from River2D 57
7.2.2 Delineation of cells 58
7.2.3 Calculation of cell habitat variables 59
7.2.4 Calculation of cell hydraulic variables 62
7.2.5 Estimation of shear stress and scour parameter values 62
7.3 Example: Upper Isolation Site 62
8 Model Research and Development Needs 66
8.1 Objectives and methods 66
8.2 Improvements to the Clear Creek application 66
8.2.1 Model additional reaches 66
8.2.2 Model additional Chinook races 67
8.2.3 Test and evaluate juvenile habitat selection 67
8.2.4 Test and calibrate redd scour predictions 67
8.3 Short-term general improvements to *inSALMO* 68
8.3.1 Represent spawner selection among reaches 68
8.3.2 Model effects of spawning gravel quality 68
8.3.3 Model differences between wild and hatchery spawners 68
8.4 Longer-term field- and model-based research 68

8.4.1 Improve representation of fry habitat selection and outmigration 68
8.4.2 Test outmigration decision submodel 69
8.4.3 Model steelhead 69
8.4.4 Evaluate the relation between cover and predation risk 70
8.4.5 Model survival and rearing in the delta 70
9 References cited 72
Appendix A: Model Description A-1
Appendix B: Software Guide B-1
Appendix C: Graphical User Interface Guide C-1
Appendix D: Limiting Factors Tool Guide D-1

List of Figures

Figure 1. Distributions of daily turbidity values, by month. Values are daily turbidity inputs for water years 2003-7. 10

Figure 2. Relation between flow and site-average dimensionless shear stress, and predicted fraction of bed that scours, for Clear Creek site 3A. Circles represent shear stress from the hydraulic model, and the curve is the power regression relationship. Squares represent the model's prediction of the fraction of the site with scouring or sediment deposition greater than 20 cm (right axis)....12

Figure 3. Relation between flow and site-average dimensionless shear stress, and predicted fraction of bed that scours, for Clear Creek site 3C. Format is the same as Figure 2. 12

Figure 4. Sensitivity analysis results for drift food availability (parameter *habDriftConc*, g drift per cm^3). Top panel: outmigrant timing, daily number of outmigrants for the three-years run, with separate lines for the eight values simulated. Lower panels: total number of live outmigrants (left) and outmigrants greater than 5 cm length (right) vs. drift availability. 20

Figure 5. Sensitivity analysis results for drift food availability (parameter *habSearchProd*, g search food per day per cm^2). Format similar to Figure 4. 21

Figure 6. Sensitivity analysis results for fish predation risk (parameter *mortFishAqPredMin*; higher values represent lower risk). Format similar to Figure 4....22

Figure 7. Sensitivity analysis results for terrestrial predation risk (parameter *mortFishTerrPredMin*; higher values represent lower risk). Format similar to Figure 4. 23

Figure 8. Sensitivity analysis results for fish movement distance (parameter *fishMoveDistParamA*; higher values let small fish select habitat over larger distances). Format similar to Figure 4. 24

Figure 9. Sensitivity analysis results for redd defense area (parameter *fishSpawnDefenseArea*, cm^2). Format similar to Figure 4. 25

Figure 10. Sensitivity analysis results for outmigration success function (parameter *fishOutmigrateSuccessL1*, the length in cm at which fish assume relative outmigration success is 0.1). Format similar to Figure 4. 26

Figure 11. Sensitivity analysis results for spawner abundance (number of spawners, of both sexes, at each of the two sites). Format similar to Figure 4....27

Figure 12. Response of egg survival and superimposition mortality to spawner abundance. The symbols represent the total number of eggs spawned, the number killed via superimposition, and the number surviving to emergence. Other sources of egg mortality were minor. 28

Figure 13. Sensitivity analysis results for number of reaches. Format similar to Figure 4 except that y axis is the number of outmigrants divided by the number of reaches in each scenario. 29

Figure 14. Outmigration timing results for water years 2008-9....33

Figure 15. Outmigrant size calibration and validation results. The x axis is the water year week (week 1 starts October 1; week 10 starts December 3). The y axis is mean length (mm) of outmigrating salmon. For four example years (2000, 2003, 2004, 2008), rotary screw trap data are compared to simulation results with four values of *outmigrantSuccessL1*....34

Figure 16. Egg survival (percent of eggs surviving to emerge) in the habitat restoration analysis, by year and site. 37

Figure 17. Fate of eggs, comparing sites 3A and 3C: percent of eggs lost to dewatering, scouring, low and high temperatures, and superimposition; and percent that emerged as fry. 38

Figure 18. Number of large (length > 5 cm) outmigrants in the habitat restoration analysis, by water year and site. 39

Figure 19. Causes of juvenile mortality in the habitat restoration analysis. Values are the percent of all mortalities due to each source. (The total number of fish dying as juveniles was much higher at 3A because more fish remained in the site instead of migrating out.) 39

Figure 20. Distributions of cell velocity and depth for juveniles, on days when they did not decide to migrate downstream. 40

Figure 21. Distribution of cell distance to hiding cover used by juveniles, on days when they did not decide to migrate downstream. 40

Figure 22. Differences between sites in juvenile fitness variables. Left: daily net energy intake, the difference between energy intake from food and expenditure on metabolism and swimming; fish only grow if this variable remains above zero. Right: daily probability of surviving mortality sources other than starvation. Extremely low survival probability values (e.g., < 0.8) typically result from occupying cells with velocity above the fish's maximum sustainable swimming speed. 41

Figure 23. Differences in flow regime between (a) water years 1986-1994 ("Before") and (b) water years 2002-2010 ("After"). The box-and-whisker plots represent the distribution of daily mean flows, for months February-May, for the before and after periods. 42

Figure 24. Differences in temperature regime between water years 1986-1994 and 2002-2010; format is the same as that of Figure 23. 43

Figure 25. October temperature distributions in before and after periods. 44

Figure 26. Number of large (length > 5 cm) outmigrants produced before and after the instream flow increase, by site. Error bars indicate 1 standard deviation over the 5 replicate simulations. 44

Figure 27. Outmigration timing, before and after the increase in instream flows. The X axis is the day of the calendar year (1 = January 1; 90 = March 31), and the Y values are the average number of outmigrants per day, over all replicates of all simulated years. The left Y axis refers to total outmigrants (black lines), and the right axis refers to large outmigrants (grey lines). 45

Figure 28. Temperatures representing the two climate change scenarios. Values are the mean of daily temperatures, by month, over the 1999–2003 period simulated. 46

Figure 29. Egg fates in the OCAP climate change analysis. Bar heights are the mean number of eggs over five replicates, using the total over the five years simulated. Error bars represent the standard deviation over the five replicates. "High temperature" and "Superimposition" are the number of eggs dying of these causes; "Emerged" is the number of eggs surviving to emergence. 47

Figure 30. Outmigration dates in the climate change analysis. The x axis is the day of the year (month/day) and the y axis is the total number of outmigrants over five replicates of five-year simulations. 47

Figure 31. LFT results for base flow. The x axis is the scaled value of base flow: x = 0.0 is the lowest base flow scenario, x = 1.0 is the highest flow scenario, and intermediate x values represent the scenario's flow as a fraction of the difference between lowest and highest. The left y axis indicates the total number of outmigrants per simulation, marked by the diamonds. The right y axis indicates the number of big outmigrants (length > 5 cm), represented by the square markers. Regression lines are shown with the regression equation and R^2 value. The multiple points for each x value occur because each limiting factors scenario is executed for eight parameter combinations that represent the model's parameter uncertainty. 51

Figure 32. LFT results for food availability. Format is the same as in Figure 28. 52

Figure 33. LFT results for hiding cover availability. Format is the same as in Figure 28. 52

Figure 34. LFT results for piscivory risk. Format is the same as in Figure 28. The x axis represents *survival* of piscivory, with higher values representing lower risk. 53

Figure 35. LFT results for redd scour. Format is the same as in Figure 28. The x axis represents redd depth; higher values represent lower risk of redd scour. 53

Figure 36. LFT results for spawning gravel availability. Format is the same as in Figure 28. ..54

Figure 37. LFT results for summer temperature. Format is the same as in Figure 28. 54

Figure 38. LFT results for winter temperature. Format is the same as in Figure 28. 55

Figure 39. LFT results for velocity shelter availability. Format is the same as in Figure 28. 55

Figure 40. LFT results for number of spawners. Format is the same as in Figure 28. 56

Figure 41. The Upper Isolation site in *inSALMO*. Cells are shaded by depth, with darker cells deeper. Black rectangles are adults, open ovals are redds. 63

Figure 42. Habitat variable distributions at the Upper Isolation site compared to sites 3A and 3C. The plots show the distributions among cells in velocity shelter availability, distance to hiding cover, and spawning gravel availability. Values are not weighted by cell area, and all cells (including those dry at normal flows) are included. 64

Figure 43. Simulated number of large (length > 5 cm) outmigrants for Upper Isolation site ("Clear Creek-UI") in comparison to sites 3A and 3C. Scenarios 1-5 refer to water years 2005-2009. On average over all these years, Upper Isolation produced 58% as many large outmigrants as 3A and almost 9 times as many as site 3C. 65

List of Tables

Table 1. Spawner initialization input for Clear Creek sites 3A and 3C. 13

Table 2. Standard parameter values for Clear Creek fall Chinook. 14

Table 3. Observed and simulated redd locations for December of calendar years 2007 (top) through 2009 (bottom), at site 3A. 31

Table 4. Observed and simulated redd locations for December of calendar years 2007 through 2009, at site 3A. 32

Table 5. Total outmigrant numbers and percent contribution by outmigrants greater than 5 cm fork length for the USFWS screw trap and Clear Creek sites 3A and 3C. Each of these sites constitutes 7% of the entire spawning habitat, so simulation results (columns 3 and 4 for total outmigrants; columns 6 and 7 for large outmigrants) are compared to 7% of screw trap estimates (columns 2 and 5). 35

Table 6. Limiting factors analysis statistical results 50

1 Introduction

1.1 Document objectives

This draft report is the primary documentation of work conducted under the project "Improvement of Salmon Life-Cycle Framework Model (*inSALMO*)", conducted under US Bureau of Reclamation contract R09PS20027 with Lang Railsback & Associates and the USDA Forest Service's Redwood Sciences Laboratory. This project started in September, 2009 and continues through September, 2011. (The project's scope of work also includes several optional tasks which are not considered in this document.)

The main objectives of this report are to:

- Describe the project's objectives;
- Document the new version of *inSALMO* produced by this project;
- Document the model's software and graphical interface;
- Describe and document an example application of *inSALMO* and development of study sites, using two sites on Clear Creek; and
- Illustrate the model's use through an example assessment of management scenarios for Clear Creek.

1.2 *inSALMO* background

This project builds on several preceding projects with the overall objective of developing a management model for Central Valley Chinook salmon populations. As the project's Statement of Work says:

> This model was originally outlined by Kimmerer and Jones & Stokes Associates, Inc. (1999). In 1999-2000, the detailed formulation was developed collaboratively by Wim Kimmerer; Warren Shaul and others at Jones & Stokes Associates, Inc.; and Steve Railsback. A draft of the model's formulation and software was submitted to FWS in 2000, when the project's initial funding expired. Additional funding was provided in 2004. Under this renewed funding, revisions to the model formulation were made by Lang, Railsback & Associates to produce the model described in this report. The software (documented separately) was modified to fully implement the revised model formulation and improve usability.

In the work funded in 2004, the Central Valley Chinook IBM was dubbed *inSALMO* version 0.5. One deliverable of that project was a statement of future information and research needs (Railsback 2005). This statement concluded that the original concept of a single management model for all parts of the Chinook life cycle has a fundamental problem: because salmon have a very complex life cycle and occupy many habitats, a model that tried to capture the entire life cycle in sufficient detail to solve many kinds of management problems would be extremely large and too complex to calibrate and use. Conversely, version 0.5 of *inSALMO* was kept simple and, consequently, lacked the detail needed to solve management problems. The original concept of a "virtual salmon population" of the Sacramento River basin that could be used to evaluate all kinds of management actions does not appear feasible. Instead, Railsback (2005) recommended that *inSALMO* be envisioned as a framework inside of which detailed models of specific, limited management problems are built.

The current project that this report describes was initiated in 2009 to pursue this new view of *inSALMO* as a general framework within which to focus on specific problems of specific salmon life stages and habitats.

A second individual-based salmonid model has been developed concurrently with *inSALMO*. The *inSTREAM* model was designed as a full-life-cycle model of resident trout communities and how their populations are affected by habitat (stream flow, temperature, turbidity; channel shape and cover availability) and biological conditions (e.g., food production, predation intensity, interspecies competition). *inSTREAM* has been thoroughly documented (Railsback et al. 2001; 2009), its fundamental assumptions have been tested and published (Railsback et al. 1999, 2005, 2006; Railsback and Harvey 2002), and it has been applied to a variety of sites and management issues (e.g., Harvey and Railsback 2007; Railsback et al. 2006; Railsback et al. 2003). A key difference between *inSTREAM* and previous versions of *inSALMO* is that *inSTREAM* uses a two-dimensional, detailed depiction of stream habitat and how it affects fish survival and growth. Adapting the two-dimensional methods of *inSTREAM* to *inSALMO* is a major element of this project.

The new version of *inSALMO* developed in this project is designated version 1.0 because it is a complete release, with tested software and complete documentation, ready for application to real salmon management questions.

1.3 Project objectives

The goal of this project, as implied in its Statement of Work, is to produce a version of *inSALMO* that is a useful part of the Ecological/Water Systems Operations Models Program (EWSOMP) authorized by the Central Valley Project Improvement Act. This new version follows the recommendations of Railsback (2005) by focusing on some specific management problems affecting a limited range of salmon life stages and habitats. Specifically, the new version is to address the freshwater life stages from spawning through smolt out-migration and these management issues (quoting the project's Statement of Work):

> 1. Seasonal flow/temp ops decisions - seasonal reservoir operation decisions are difficult to make for salmonids to balance flows and temperature regimes with the needs of the fishery? The addition of this tool will allow managers to evaluate different flow scenarios and their impacts on the fishery. The life stages that managers focus on are the most sensitive life stages to flow and temperature: eggs, alevins, and fry.
>
> 2. Annual planning to best use environmental water- we have fixed amount of water, how and when do we use it over the course of a year to produce the most fish or the fish in the best condition out of the rivers? The addition of this tool will allow managers to evaluate different uses of environmental water to maximize the beneficial use of environmental water on CVP streams. To make a comprehensive assessment we would need to look at all life stages.
>
> 3. Restoration use- planning, design and evaluation of restoration activities. This expanded capability will be used by FWS to facilitate restoration decision making by comparing different restoration scenarios and evaluating the effects of restoration projects. Many restoration activities are for: spawning, egg incubation, and juvenile rearing habitat.
>
> 4. Limiting factors analysis- the addition of this tool will allow managers to consider different management scenarios to evaluate which actions have the greatest affects on the fishery.

The project has eight specific tasks.

Task 1: Upgrade software interface. Version 0.5 of *inSALMO* was controlled and operated by editing a complex set of text files. This task is to make the model easier to use via improvements such as making installation easier; make it easier to apply the model to new sites; pro-

vide a graphical interface to set up, run, and examine routine model runs; and making input and output easier to view and edit in widely used software tools such as Excel.

Task 2: Incorporate two-dimensional flow and habitat modeling. Previous versions of *inSALMO* used a one-dimensional depiction of stream habitat, coarsely depicting habitat only at the scale of kilometer-long reaches. The model was modified by inserting the two-dimensional habitat representation, and the ways that fish select and respond to habitat at a small scale (cells of several square meters), from *inSTREAM*, a related instream flow model for resident trout (Railsback et al. 2009).

Task 3: Upgrade interface for temperature and flow data. This task is to closely link *inSALMO*'s time-series inputs of temperature and flow to values produced by other Central Valley operations models.

Task 4: Testing and evaluation. This task calls for evaluation of the revised *inSALMO* by applying it to example management issues at a specific study area. Several subtasks are to develop input for several reaches of Clear Creek, conduct an example decision-support application, and to integrate the model with existing fisheries agency studies and management.

Task 5: Training sessions and documentation. The ability to apply and use *inSALMO* is to be transferred to federal agency staff working on Central Valley salmon management.

Task 6: Identify future modeling needs. A document addressing further improvements to the model and its application is to be developed. This document is incorporated in this report as Section 5.3.

Task 7: Limiting factors analysis. This task is to provide tools for evaluating various factors that affect freshwater life stages of salmon (e.g., flow timing and magnitude, temperature, predators, habitat condition) by their relative effects on smolt production.

Task 8: Progress reports and meetings. This task simply funds monthly reports and progress meetings.

1.4 Document overview

All the written products of this project are consolidated in this document.

Section 2 provides an overview description of *inSALMO* 1.0, with the complete model description at Appendix A. The complete model description is also distributed with the model software as a help file.

Section 3 describes how to install and run the model. It is supplemented by two appendices that are also included as help files packed with the model software. Appendix B is a detailed guide to the model's software, including input files, outputs, the Experiment Manager; and it includes a troubleshooting guide listing common problems. Appendix C is a guide to the Graphical User Interface (GUI), an optional new software tool for setting up and controlling model runs. This section documents the products of tasks 1, 2, and 3.

The application of *inSALMO* to Clear Creek under Task 4 is described in Section 4. This section describes the modeled reaches, the input used to model those reaches, and how the model was calibrated and analyzed. The application not only documents this example application but serves as a template for applying the model to additional sites.

The example decision-support application also part of Task 4 is documented in Section 5. The section uses the model for three applications: evaluating the benefits of habitat restoration,

contrasting two alternative flow and temperature regimes, and examining the effects of two climate change scenarios affecting water temperature.

Section 6 illustrates use of the Limiting Factors Tool, a software tool built into the GUI to automate analysis of which factors have strong effects on salmon reproductive success. The tool is fully documented at Appendix D. This section and appendix are products of Task 7.

One product of Task 4 is at Section 7: a way of integrating use of *inSALMO* with agency studies. Methods for using data from US Fish and Wildlife Service instream flow studies to create new model reaches are described and illustrated.

The final Section 8 provides the product of Task 6. It reports future research and development needs, remaining improvements that appear of high priority to the usefulness of *inSALMO*.

1.5 Acknowledgements and credits

This project was funded by the US Bureau of Reclamation's Mid-Pacific Region, Claire Hsu, project manager. The US Fish and Wildlife Service was represented by Dan Cox, Mark Gard, and Julie Zimmerman.

Previous versions of *inSALMO* and related models have been funded by US Fish and Wildlife Service, USDA Forest Service, US Environmental Protection Agency, the Electric Power Research Institute, Western Area Power Administration, Pacific Gas and Electric, and Southern California Edison. Much of this predecessor work was conducted through the Department of Mathematics at Humboldt State University; information is at: www.humboldt.edu/ecomodel.

This project was conducted by an interdisciplinary team lead by Lang, Railsback & Associates (LRA; www.LangRailsback.com) and the USDA Forest Service, Pacific Southwest Research Station, Redwood Sciences Laboratory (RSL). LRA and RSL collaborated via Forest Service Agreement No. 10-CO-11272133-001, with Steve Railsback and Bret Harvey as project managers. Bret Harvey (RSL) directed the application of *inSALMO* to Clear Creek, including field studies. Jason White and Rod Nakamoto of RSL assisted with field studies, data interpretation and input development, and numerous model analyses.

The *inSALMO* software was originally developed by Steve Jackson (Jackson Scientific Programming, McKinleyville CA). Colin Sheppard (Arcata, CA) developed the graphical user interface and currently maintains the model code. Charles Sharpsteen (Arcata, CA) provided the software expertise to overcome a significant obstacle, producing Windows executables of Swarm models.

Jeff Anderson, Corin Pilkington, and Bonnie Pryor of Northern Hydrology (McKinleyville CA) provided geomorphic and hydraulic modeling expertise and GIS support.

We thank the many resource agency staff who devoted substantial amounts of time to this project. These included Claire Hsu, Dan Cox, Mark Gard, Matt Brown, Derek Hilts, Craig Anderson, Steve Barre, Robert Chase, James Earley, John Hannon, Li-Ming He, Cathy Marcinkevage, Jim Smith, David Swank, and Julie Zimmerman.

2 Model Description

A complete description of version 1.0 of *inSALMO* is provided at Appendix A. The model description is also distributed with the model's software as a help file. The model description was adapted from the description by Railsback et al. (2009) of *inSTREAM* version 4.2.

A key characteristic of *inSALMO* 1.0 is that it models only the spawning, incubation, and stream rearing parts of the salmon life cycle, instead of modeling the entire life cycle as version 0.5 did. This change is to allow focus on issues of the freshwater life stages in sufficient detail to address important management questions (Section 1.3) without the cost in complexity and uncertainty of representing other life stages.

The model description appendix starts with an overview of the model's basic assumptions and elements, and then explains terminology used in the model. It then provides complete detail on the characteristics and methods of entities in the model: habitat reaches and cells; and salmon spawners, redds, and juveniles. It ends with a summary of the model's actions and how they are scheduled for execution. Extensive citations provide justification for the model's assumptions.

3 Software Installation and Use

This section provides basic information for installing and executing the *inSALMO* 1.0 software. Additional detail is provided in two appendices that are also available as help files from the model's graphical user interface. Appendix B is a detailed guide to the model software, describing:

- The contents and formats of all the input files,
- The graphical interfaces available when the model executes,
- Standard and optional output files, and
- Operation of the Experiment Manager tool for automated simulation experiments.

The GUI is documented separately as Appendix C. New users can install and run *inSALMO* through the GUI productively, using only the information in this GUI guide. It is helpful and recommended, but not essential, that people who use *inSALMO* via the GUI also be familiar with the input and output files described in the Appendix B software guide; and the software guide will be helpful for identifying problems that can arise when using the GUI.

3.1 Installing *inSALMO*

The *inSALMO* software is distributed for use on Windows computers as a zip archive file. The zip file contains the model's executable software; the Swarm libraries that *inSALMO* uses; the graphical user interface (GUI), an optional way to edit input and run the model; a set of example input; and the source code to *inSALMO*.

Installing the model is a simple two-step process of extracting the zip file and then creating a shortcut file so the GUI can be started from the desktop. It should not require any administrative privileges because the installation does not affect the Windows registry. This process is documented in Section 3.2 of the GUI guide.

3.2 Using the Graphical User Interface

The GUI for *inSALMO* can be used to create and modify "projects" or applications of the model to a particular site and question, and then execute the model and begin the results analysis process. The most useful parts of the GUI are tools for (a) setting up and executing simulation experiments in which one or several model inputs (parameter values, flow or temperature input files, etc.) are varied; and (b) executing "limiting factors" experiments that evaluate the relative effect on spawning and rearing success of factors such as flow, temperature, spawning gravel, and in-stream cover.

The GUI is executed simply by clicking on the *inSALMO* shortcut placed on the computer desktop during installation. (The software cannot be executed through the Windows "Start" menu.) Instructions in the GUI guide can then be used to set up, run, and examine simulation experiments.

3.3 Using *inSALMO* without the GUI

Some users may prefer using *inSALMO* from the Windows Explorer or command line instead of through the GUI. The following steps can be used. These instructions refer to the directory tree installed when the *inSALMO* zip file is extracted; the location of this tree depends on where it was extracted. Here, the directory `\inSALMO_1.0` refers to the top level of the *inSALMO* directory tree; it may actually be located somewhere such as `C:\Documents and Settings\(user name)\My Documents\inSALMO_1.0`.

1) Create a new working project directory that contains the model inputs you want to modify and use. The easiest way to do this is to copy the default input packaged with

the software. Look for a "default" directory under `\inSALMO_1.0\Projects` (e.g., `\Projects\ClearCreekDefault`). Copy this directory to any location, though keeping the new directory under `\Projects` will make it easier to use the GUI if you later choose to.

2) Create a Windows batch file that executes the model in graphics mode and place it in the project directory. It should be named something like `insalmo.bat`. You can execute the model, using the input files in your current project directory, by double-clicking on this file from Windows Explorer. The batch file should contain these lines in plain text (but replace both instances of `\Documents and Settings\(user name)\Desktop\inSALMO_1-0` with the actual path to the top-level *inSALMO* directory on your computer; the double-quote marks are important):

```
set SWARMHOME="C:\Documents and Settings\(user name)\Desktop\inSALMO_1-0\Code\Swarm"
set path=%SWARMHOME%\bin;%PATH%
"C:\Documents and Settings\(user name)\My Documents\inSALMO_1-0\Code\inSALMO\.libs\insalmo.exe"
```

3) Your project directory should also include a Windows batch file that executes the model in batch (non-graphics) mode. This file should be named something like `insalmo-batch.bat`. It should be identical to `insalmo.bat` except that in the final line replace `\insalmo.exe` with `\insalmo.exe -b`.
4) You can access the help files by double-clicking on them from Windows Explorer; they are in the directory `\inSALMO_1.0\Help`. The files should open in the same searchable, indexed format that they do from the GUI. (The help files open in a Windows program called Windows HTML Help and are in a format called Compiled HTML Help.)

You can now edit and execute the project using the methods and information provided in the software guide (Appendix B).

3.4 Modifying and compiling the source code

Occasionally, users may choose to update the *inSALMO* program by recompiling it instead of obtaining a new executable from the program developers. For example, the software guide provides instructions for changing which species are modeled or the size classes used for output. The developers may also provide updates to *inSALMO* that are best installed by recompiling it.

The source code is included in the model distribution at `\inSALMO_1.0\inSALMO`. The following steps are necessary to recompile the code (the first step, installing MinGW, is only needed once.)

1) Install the MinGW compiler and its MSys terminal window software, and the Swarm libraries. Follow the directions on installing MinGW and Swarm at this URL: http://www.swarm.org/index.php/Swarm_and_MinGW#Installing_Swarm_and_MinGW_to_compile_models (note that this process requires putting a copy of the Swarm libraries at `C:\swarm`).
2) You should now have a new program called `MinGW Shell` (also called `Msys`) installed on your computer. This is a Unix terminal window within which you must compile *inSALMO*.
3) Start `MinGW Shell` and type these commands:

```
cd /c/.../inSALMO_1.0/inSALMO

make clean

make
```

(Here, `.../inSALMO_1.0/inSALMO` refers to the directory with the code; `/c/` refers to the C drive. Spaces in the directory name must be preceded by a backslash. An example is: `cd /c/Documents\ Settings/My\ Username/My\ Documents/inSALMO_1-0/Code/inSALMO/`)

4) Exit `MinGW Shell`.

The newly recompiled *inSALMO* executable will now be used automatically in all project directories. (There are ways, not explained here, to use different versions of the executable in different projects.)

4 Example Application to Clear Creek

Task 4 of this project calls for a test application and evaluation of *inSALMO* 1.0, with three objectives defined as subtasks. Subtask 4.1 is to develop the study site, which includes collecting and assembling the data needed to apply the model. Clear Creek was selected early in the project as the test application site, due primarily to its manageable size, and because recent restoration activities have produced a great deal of information and the ability to contrast sites with distinctly different habitat characteristics.

This section describes how the model was applied to the Clear Creek application, from preparation of site-specific input and parameter values through calibration to data from the site. It serves not only as documentation of Task 4.1 work, but also as a guide, by example, for applying *inSALMO* to other study sites.

4.1 Study site selection

Selecting specific reaches to model was the first step of the example application. Given Clear Creek's width and habitat diversity, reach lengths of about 500 m were assumed provide a good compromise between capturing the site characteristics and computational burden. Criteria for study sites included (a) each site having general habitat characteristics useful for representing large and important parts of Clear Creek, (b) having differences between sites that can represent management alternatives concerning habitat restoration, and (c) the availability of data for model development and evaluation.

The project team agreed on two study sites that meet these criteria well. The two sites are in the lower alluvial segment of Clear Creek, where restoration of gravel mining effects and other disturbance has been extensive but not complete. The US Fish and Wildlife Service (USFWS 2008) provides a detailed description of the study area and a thorough overview of restoration projects in Clear Creek. The model sites, lying between river km 4 and river km 6, are sub-sections of USFWS-designated areas "3A" and "3C" and retain those names. Restoration projects undertaken in the 3A reach include a major channel re-alignment, placement of instream structures and injection of spawning gravel. The resulting series of riffles and pools differs markedly from Reach 3C, where restoration work has not yet begun, although Reach 3C has been affected by upstream gravel augmentation. Reach 3C remains predominantly a deeply-incised, fast-flowing run, referred to by the USFWS as "The Chute".

The portion of 3A modeled in *inSALMO* has a thalweg length of approximately 490 m, and the 3C model reach is approximately 460 m long.

4.2 Habitat input and parameters

This subsection describes how habitat input for inSALMO was assembled

4.2.1 Time series inputs: flow, temperature, and turbidity

Flow at both study sites was represented by the USGS gage at Igo (USGS 11372000). This gage measures releases from Whiskeytown Reservoir and some but not all tributary inflow between the reservoir and study sites.

For daily mean temperature input, we used the water temperature record from a nearby USFWS monitoring site (Restoration Grove) for the period 4 March 2001 to 31 December 2008. To estimate water temperature at the study sites before development of the extensive Clear Creek water temperature monitoring program of the USFWS, we modeled water temperature using streamflow and Redding air temperature as predictor variables. For 2001 to 2008, daily mean air temperature and the log of streamflow predicted daily mean water temperature with $R^2 = 0.87$.

Graham Matthews & Associates provided turbidity monitoring data for water years 2003 - 2007 from a station about 500 m upstream of Site 3A. These data allowed computation of daily mean turbidity for 534 days. Clear Creek streamflow predicted turbidity for these days with R^2 = 0.76. We used this relationship to estimate turbidity when direct measurements were unavailable.

Flow and temperature distributions under recent instream flow management policies are described and illustrated graphically in Section 5.2. Turbidity input is generally low (Figure 1), below 5 NTU (the threshold below which it is assumed in *inSALMO* to have no effect on feeding) on more than 91% of days.

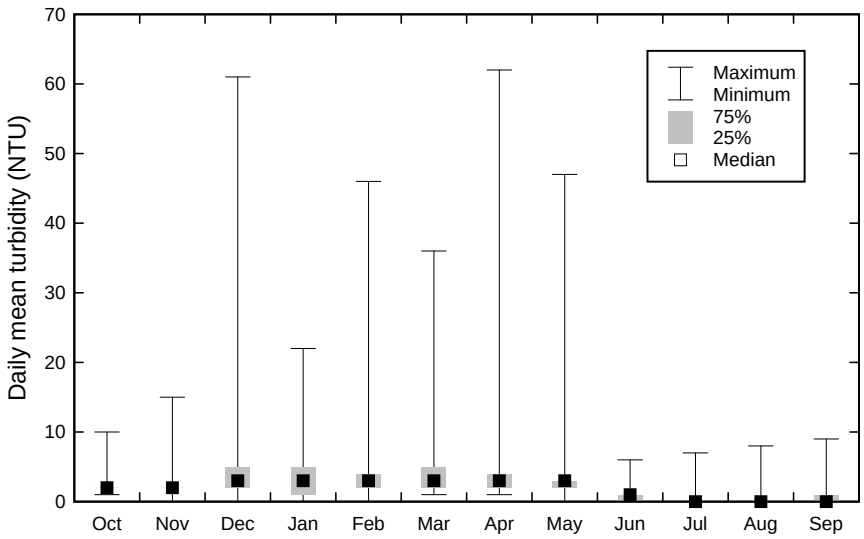


Figure 1. Distributions of daily turbidity values, by month. Values are daily turbidity inputs for water years 2003-7.

4.2.2 Cell geometry

Our geometric representation of the study sites was based on the meshes used by Mark Gard of the U.S. Fish and Wildlife Service (USFWS) Sacramento Fish and Wildlife Office in his hydraulic models. His hydraulic modeling used the River2D model (www.river2d.ualberta.ca). The boundaries of the inSALMO sites are the same as those of the hydraulic models. However, the meshes used in River2D were finer than desirable for inSALMO so we developed new cells that are generally larger than those in River2D. Our new cell geometry was based on bed topography and field observations of habitat (including our cover variables as well as depth and velocity). The cell boundaries were manually placed using a geographic information system (GIS). Starting with aerial photography of the site overlaid with the River2D geometry and field habitat observations, cell corners were selected to capture the important variation in habitat while producing no more cells than necessary. Consequently, cells tend to be small where habitat varies sharply with distance (especially along banks) and large in more homogeneous areas. This process produced 552 cells at site 3A and 162 cells at 3C.

4.2.3 Hydraulics

Hydraulic input provides the depth and velocity in each cell, at a number of flows covering the range occurring in the flow input. These inputs are used by *inSALMO* each simulated day to interpolate depth and velocity from daily flow. Mark Gard provided depths and velocities from River2D simulations at 23 flows within the calibrated range of the hydraulic model (from 1.4 to 25.5 m^3/s; 50 to 900 cfs). To cover high-flow events, he also provided results extrapolated

above the calibrated range, for four flood flows from 56.6 to 1416 m^3/s (2000 to 50,000 cfs).

These hydraulic model results were averaged from the more-detailed River2D mesh into our larger cells, via GIS. For each *inSALMO* cell, values from all River2D points inside the cell were averaged to estimate the cell's depth and velocity. This process was repeated for each of the 27 flows simulated in River2D.

The high-flow River2D simulations at site 3A were found to produce unrealistic results. Perhaps because the River2D model includes little of the floodplain, it produced depths that appeared unrealistically high; consequently, the simulated velocities at flood flows appeared unrealistically low. We sidestepped this issue by including, for site 3A, depths and velocities from only the highest of the flood flows instead of from all four. Consequently, for any flow between 25.5 and 1416 m^3/s (900 to 50,000 cfs) *inSALMO* linearly interpolates cell depths and velocities between the values for 25.5 and 1416 m^3/s.

4.2.4 Shear stress parameters

The formulation for scouring mortality of redds uses the probabilistic model of Haschenburger (1999), which assumes that the probability of a redd being scoured during a high flow event increases with the dimensionless shear (Shields) stress, and that shear stress increases with flow according to a power relationship defined by the habitat parameters *habShearParamA* and *habShearParamB*. These parameters were evaluated via logarithmic regression on shear stress values estimated from hydraulic simulations at a number of high flows. The probability of a redd being scoured also decreases with redd depth; the parameter *mortReddScourDepth* (cm) is the assumed depth to which eggs are buried. We set this parameter to 20 cm.

For site 3A, the uncalibrated River2D simulations for high flows (Sect. 4.2.3) were inadequate to estimate shear stress at high flows. Instead we obtained shear stress estimates from the one-dimensional hydraulic model analysis originally used to design the 3A site restoration. These estimates were provided by Jeff Anderson (Northern Hydrology & Engineering, McKinleyville CA, June 14, 2011), using the HEC-RAS model with input from topographic surveys of the 3A reach made in approximately 2005. Shear stress was calculated for eight transects in, and just above and below, the study site, for flows with recurrence intervals from 1 to 500 years. These calculations were made to represent stress in the channel only, not in the floodplain. The mean shear stress was calculated by averaging the values from all eight transects. A median particle diameter of 0.035 m was determined as a sitewide average from pebble counts conducted at the site, as reported in Table 3-1 of GMA (2007). The resulting relationship (Figure 2) provides parameter values of: *habShearParamA* = 0.013 and *habShearParamB* = 0.40. The scour model then predicts the onset of scouring at about 100 m^3/s and widespread scour (>25% of the bed) during flow events with peaks above 250 m^3/s (Figure 2).

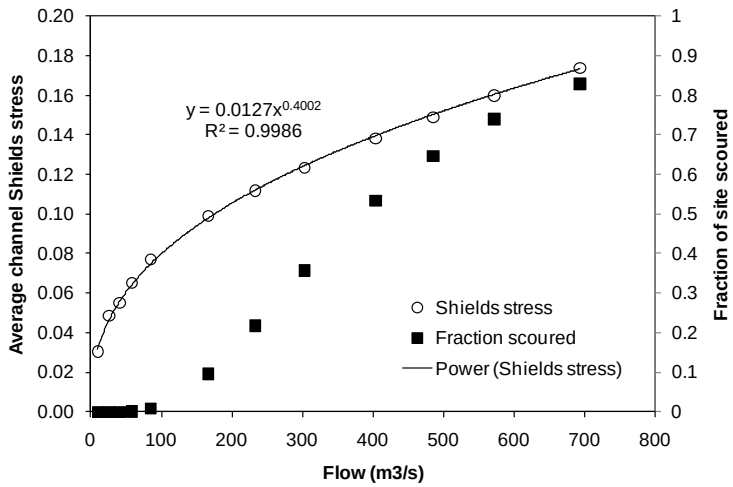


Figure 2. Relation between flow and site-average dimensionless shear stress, and predicted fraction of bed that scours, for Clear Creek site 3A. Circles represent shear stress from the hydraulic model, and the curve is the power regression relationship. Squares represent the model's prediction of the fraction of the site with scouring or sediment deposition greater than 20 cm (right axis).

For site 3C, the two-dimensional River2D simulations at high flow provided an adequate model of shear stress. The relationship between shear stress and flow from those simulations produced parameter values of: *habShearParamA* = 0.010 and *habShearParamB* = 0.49. The model then predicts scouring to begin at flows well below 100 m^3/s and become widespread (>25% of the bed) during flows around 150 m^3/s (Figure 3). The onset of scour at lower flows at site 3C, compared to 3A, is expected because the habitat restoration at 3A was designed to provide a floodplain that allows the stream to widen and reduce shear stress at flows (80-100 m^3/s) that previously filled the banks.

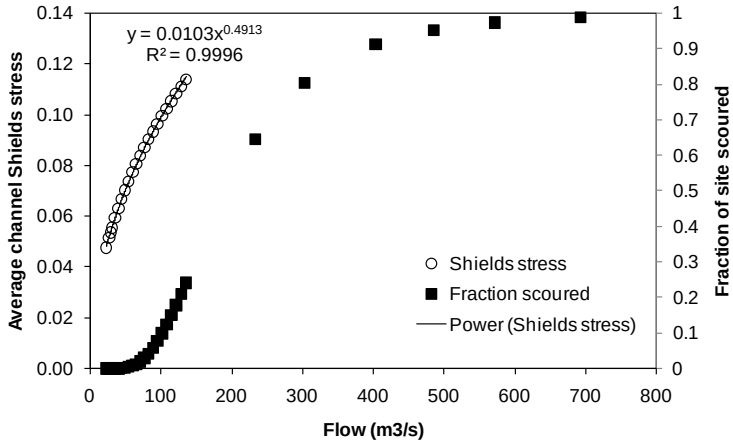


Figure 3. Relation between flow and site-average dimensionless shear stress, and predicted fraction of bed that scours, for Clear Creek site 3C. Format is the same as Figure 2.

4.3 Salmon input and parameter values

This section describes the model input for fall Chinook salmon at the Clear Creek sites. The input includes spawner initialization data and fish parameter values.

4.3.1 Spawner initialization data

Spawner initialization data specifies the number, characteristics, and timing of adult salmon arriving to spawn. This input was developed for Clear Creek from data collected and published by the fisheries agencies (Table 1). Estimates of adult numbers for each site rely on annual California Department of Fish and Game (CDF&G) spawning escapement estimates for an approximately 6.75 km section of lower Clear Creek (source: Data query from CalFish.org website). The CDF&G survey reach includes both the 3A and 3C model reaches. Estimates for the 3A and 3C reaches were produced by multiplying the overall reach estimate by the proportion of the overall reach length represented by each model site (~7% for each site). Estimates of adult sizes and sex ratio were generated from unpublished carcass data provided by CDF&G.

Table 1. Spawner initialization input for Clear Creek sites 3A and 3C.

Year	Arrival reach	Number of spawners	Fraction female	Arrival start date	Arrival end date	Arrival ratio	Female length mean	Female length SD
1998	ClearCreek-3A	324	0.58	10/2/1998	10/31/1998	0.0001	85	8.1
1999	ClearCreek-3A	580	0.51	10/2/1999	10/31/1999	0.0001	85	8.1
2000	ClearCreek-3A	485	0.5	10/2/2000	10/31/2000	0.0001	85	8.1
2001	ClearCreek-3A	788	0.51	10/2/2001	10/31/2001	0.0001	85	8.1
2002	ClearCreek-3A	1165	0.51	10/2/2002	10/31/2002	0.0001	85	8.1
2003	ClearCreek-3A	687	0.57	10/2/2003	10/31/2003	0.0001	85	8.1
2004	ClearCreek-3A	461	0.58	10/2/2004	10/31/2004	0.0001	85	8.1
2005	ClearCreek-3A	1075	0.65	10/2/2005	10/31/2005	0.0001	85	8.1
2006	ClearCreek-3A	611	0.57	10/2/2006	10/31/2006	0.0001	85	8.1
2007	ClearCreek-3A	299	0.64	10/2/2007	10/31/2007	0.0001	85	8.1
2008	ClearCreek-3A	557	0.65	10/2/2008	10/31/2008	0.0001	85	8.1
2009	ClearCreek-3A	600	0.57	10/2/2009	10/31/2009	0.0001	85	8.1
1998	ClearCreek-3C	304	0.58	10/2/1998	10/31/1998	0.0001	85	8.1
1999	ClearCreek-3C	545	0.51	10/2/1999	10/31/1999	0.0001	85	8.1
2000	ClearCreek-3C	455	0.5	10/2/2000	10/31/2000	0.0001	85	8.1
2001	ClearCreek-3C	739	0.51	10/2/2001	10/31/2001	0.0001	85	8.1
2002	ClearCreek-3C	1094	0.51	10/2/2002	10/31/2002	0.0001	85	8.1
2003	ClearCreek-3C	645	0.57	10/2/2003	10/31/2003	0.0001	85	8.1
2004	ClearCreek-3C	433	0.58	10/2/2004	10/31/2004	0.0001	85	8.1
2005	ClearCreek-3C	1009	0.65	10/2/2005	10/31/2005	0.0001	85	8.1
2006	ClearCreek-3C	573	0.57	10/2/2006	10/31/2006	0.0001	85	8.1
2007	ClearCreek-3C	281	0.64	10/2/2007	10/31/2007	0.0001	85	8.1
2008	ClearCreek-3C	522	0.65	10/2/2008	10/31/2008	0.0001	85	8.1
2009	ClearCreek-3C	600	0.57	10/2/2009	10/31/2009	0.0001	85	8.1

Source: data from Clear Creek Reach 6, 1999 - 2001 provided by Newton and Brown (2004).

4.3.2 Fall Chinook parameter values

The following table documents all fish parameters used in the application of *inSALMO* 1.0 to the Clear Creek sites. Many parameter values are unchanged from the standard trout values established by Railsback et al. (2009) because we identified no substantial basis for different values. Values that are different from standard trout values, because they reflect important differences between Chinook salmon and trout, or characteristics of the Clear Creek sites, are explained in the third column. The third column also explains parameters in *inSALMO* but not *inSTREAM*. Fuller explanations of fish parameters and their values are in the model description (Appendix A; the help file version of the model description includes an index of parameter names).

Table 2. Standard parameter values for Clear Creek fall Chinook.

Salmon parameter	Standard value for Clear Creek	Definition and basis for value (if other than Railsback et al. 2009)
fishCaptureParam1	1.6	
fishCaptureParam9	0.5	
fishCmaxParamA	0.628	
fishCmaxParamB	-0.3	
fishCmaxTempF1	0.05	
fishCmaxTempF2	0.05	
fishCmaxTempF3	0.5	
fishCmaxTempF4	1	
fishCmaxTempF5	0.8	
fishCmaxTempF6	0	
fishCmaxTempF7	0	
fishCmaxTempT1	0	
fishCmaxTempT2	2	
fishCmaxTempT3	10	
fishCmaxTempT4	22	
fishCmaxTempT5	23	
fishCmaxTempT6	25	
fishCmaxTempT7	100	
fishDetectDistParamA	4	
fishDetectDistParamB	2	
fishEnergyDensity	5900	
fishFecundParamA	690	Multiplier in equation for number of eggs as a function of length. Source: equation for Sacramento River Chinook, Table 1 of Healy and Heard (1984), with their relation between fork length and postorbit-hypural length. Parameters are modified to use length in cm instead of mm.
fishFecundParamB	0.522	Exponent in fecundity equation. Source: same as above. Note that this parameter value is unusual because it is less than one. In most relationships (and most for Chinook; Healy and Heard 1984) fecundity increases more than linearly with length—values are typically between 1.2 and 2.5.

Salmon parameter	Standard value for Clear Creek	Definition and basis for value (if other than Railsback et al. 2009)
fishFitnessHorizon	90	
fishMaxSwimParamA	2.8	
fishMaxSwimParamB	21	
fishMaxSwimParamC	-0.0029	
fishMaxSwimParamD	0.084	
fishMaxSwimParamE	0.37	
fishMoveDistParamA	50	Estimate for relatively low-gradient mid-sized streams.
fishMoveDistParamB	2	
fishOutmigrateSuccessL1	5	Length (cm) at which expected outmigrant success is 0.1.
fishOutmigrateSuccessL9	12	Length (cm) at which expected outmigrant success is 0.9.
fishRespParamA	30	
fishRespParamB	0.784	
fishRespParamC	0.0693	
fishRespParamD	0.03	
fishSearchArea	20000	
fishSpawnEggViability	0.8	
fishShelterSpeedFrac	0.7	
fishSpawnDefenseArea	200,000	The area (cm^2) a female spawner defends around her redd. Set to four times the area of a redd, as observed in Columbia River tributaries by Burner (1951).
fishSpawnStartDate	10/1	The first date on which adults are allowed to spawn. This value lets adults spawn as soon as they arrive at the site. (The arrival period is generally the month of October.)
fishSpawnEndDate	11/30	The last date on which adults can spawn, and the date on which any unspawned adults are forced to spawn. M. Brown (email 5/25/2011) indicates that spawning in small numbers continues through November.
fishSpawnDSuitD1	0.0	Parameters fishSpawnDSuitD1 - 5 and fishSpawnDSuitS1 - 5 define spawning habitat suitability (dimensionless; 0-1) for depth (cm). Source: habitat suitability criteria for spawning in Clear Creek, from USFWS (2011).
fishSpawnDSuitD2	12	
fishSpawnDSuitD3	27	
fishSpawnDSuitD4	33.5	
fishSpawnDSuitD5	204	
fishSpawnDSuitS1	0.0	
fishSpawnDSuitS2	0.0	
fishSpawnDSuitS3	0.95	
fishSpawnDSuitS4	1.0	
fishSpawnDSuitS5	0.0	
fishSpawnMaxFlowChange	0.2	
fishSpawnMaxTemp	14	The highest temperature (C) at which salmon spawn. Estimated as the highest temperature that does not cause rapid egg mortality (see redd mortality parameters below).

Salmon parameter	Standard value for Clear Creek	Definition and basis for value (if other than Railsback et al. 2009)
fishSpawnMinTemp	5	The lowest temperature (C) at which salmon spawn. Estimated as the lowest temperature that does not cause rapid egg mortality.
fishSpawnProb	0.2	Daily probability that an otherwise-ready spawner will spawn. Value chosen to cause spawning within a few days.
fishSpawnVSuitS1	0.0	Parameters fishSpawnVSuitV1 - 5 and fishSpawnVSuitV1 - 5 define spawning habitat suitability (dimensionless; 0-1) for velocity (cm/s). Source: habitat suitability criteria for spawning in lower Clear Creek, from USFWS (2011).
fishSpawnVSuitS2	0.0	
fishSpawnVSuitS3	0.06	
fishSpawnVSuitS4	1.0	
fishSpawnVSuitS5	1.0	
fishSpawnVSuitS6	0.0	
fishSpawnVSuitV1	0.0	
fishSpawnVSuitV2	2.3	
fishSpawnVSuitV3	3.0	
fishSpawnVSuitV4	54	
fishSpawnVSuitV5	61	
fishSpawnVSuitV6	192	
fishSpawnWtLossFraction	0.4	Fraction of spawner's body mass lost upon spawning.
fishTurbidExp	-0.0711	
fishTurbidMin	0.1	
fishTurbidThreshold	5	
fishWeightParamA	0.00411	Multiplier in equation for juvenile salmon body mass (g) as a function of fork length (cm). Source: relation documented by Petrusso and Hayes (2001) for juvenile Chinook salmon in the Sacramento River, with units converted.
fishWeightParamB	3.49	Exponent in equation for juvenile salmon body mass (g) as a function of fork length (cm). Source: same as previous.
mortFishAqPredD1	20	Depth (cm) at which survival of predation by fish is reduced by 10%. The value is an estimate reflecting presence, but not dominance, of relatively small piscivores such as trout. Most piscivorous fish are assumed larger than trout.
mortFishAqPredD9	10	Depth (cm) at which survival of predation by fish is reduced by 90%. Source: same as previous.
mortFishAqPredF1	18	
mortFishAqPredF9	0	
mortFishAqPredH1	200	Distance (cm) to hiding cover at which survival of fish predation is reduced by 10%.
mortFishAqPredH9	0	Distance (cm) to hiding cover at which survival of fish predation is reduced by 90%. Value is estimated from the assumptions that cover provides some protection from large fish but even dense cover does not provide complete safety, especially from smaller predators such as trout.
mortFishAqPredL1	4	Salmon length (cm) at which survival of predation by fish is reduced by 10%. Source: estimate based on preva-

Salmon parameter	Standard value for Clear Creek	Definition and basis for value (if other than Railsback et al. 2009)
		lence of large-gaped piscivores such as pikeminnow.
mortFishAqPredL9	18	Length (cm) at which survival of predation by fish is reduced by 90%. Source: same as previous.
mortFishAqPredMin	0.90	Daily probability of surviving predation by fish, before adjustment by risk reduction factors. Source: pre-calibration estimate.
mortFishAqPredT1	16	Temperature (C) at which predation of fish is reduced by 10%. Value is an estimate reflecting dominance of warmwater predators (e.g., pikeminnow) but presence of coldwater piscivores (e.g., trout). Response of pikeminnow predation to low temperatures is evaluated from maximum consumption response to temperature reported by Vigg and Burley (1991).
mortFishAqPredT9	8	Temperature (C) at which predation of fish is reduced by 90%. Source: same as previous.
mortFishAqPredU1	5	
mortFishAqPredU9	80	
mortFishConditionK1	0.3	
mortFishConditionK9	0.6	
mortFishHiTT1	28	Temperature (C) at which daily survival of acute thermal effects is 10%. Source: estimate considering information from Myrick and Cech (2004).
mortFishHiTT9	24	Temperature (C) at which daily survival of acute thermal effects is 90%. Source: same as previous.
mortFishStrandD1	-0.3	
mortFishStrandD9	0.3	
mortFishTerrPredD1	5	Depth (cm) at which survival of terrestrial predators is increased by 10%. Source: estimate reflecting dominance of predators that are relatively good swimmers (e.g., mergansers).
mortFishTerrPredD9	200	Depth (cm) at which survival of terrestrial predators is increased by 90%. Source: same as previous.
mortFishTerrPredF1	18	
mortFishTerrPredF9	0	
mortFishTerrPredH1	500	
mortFishTerrPredH9	-100	
mortFishTerrPredL1	6	
mortFishTerrPredL9	3	
mortFishTerrPredMin	0.986	
mortFishTerrPredT1	10	
mortFishTerrPredT9	50	
mortFishTerrPredV1	20	Velocity (cm/s) at which survival of terrestrial predators is reduced by 10%. Source: estimate reflecting presence of relatively fast runs that do not always provide turbulence to conceal fish.
mortFishTerrPredV9	200	Velocity (cm/s) at which survival of terrestrial predators is reduced by 90%. Source: same as above.
mortFishVelocityV1	1.8	
mortFishVelocityV9	1.4	

Salmon parameter	Standard value for Clear Creek	Definition and basis for value (if other than Railsback et al. 2009)
mortReddDewaterSurv	0.9	
mortReddHiTT1	23	Temperature (C) at which daily survival by eggs and alevins of thermal stress and related disease is 10%. Source: estimate reproducing several patterns summarized from the literature by Myrick and Cech (2004): Temperature effects become detectable around 12°; mortality is clear but not rapid by 14°; and mortality is strong and rapid by around 16.5°.
mortReddHiTT9	17.5	Temperature (C) at which daily egg and alevin survival of thermal stress and related disease is 90%. Source: same as previous.
mortReddLoTT1	1.7	Temperature (C) at which daily egg and alevin survival of low-temperature stress is 10%. Value is estimated from information in Myrick and Cech (2004).
mortReddLoTT9	4.0	Temperature (C) at which daily egg and alevin survival of low-temperature stress is 90%. Source: same as previous.
mortReddScourDepth	20	Depth (cm) of scour that causes destruction of a redd. Value of 20 cm is supported by DeVries (1997).
reddDevelParamA	33000	Parameter A in equation relating redd development to temperature.
reddDevelParamB	-2.04	Parameter B in equation relating redd development to temperature.
reddDevelParamC	-7.58	Parameter C in equation relating redd development to temperature.
reddNewLengthMin	3.5	Minimum fork length (cm) of juvenile salmon upon emergence. Source: unpublished data provided by Michael Sparkman (California Department of Fish & Game, Arcata, CA) from four emergence traps deployed in the Redwood Creek drainage, Humboldt County, 1996-99.
reddNewLengthMax	4.1	Maximum fork length (cm) at emergence. Source: same as previous.
reddSize	56,000	Mean area (cm^2) of Chinook salmon redds (pit and tailspill). Source: Clear Creek observations reported by Newton and Brown (2004). Corresponds to values reported by Burner (1951) for Chinook in lower Columbia basin.

4.4 Sensitivity analysis

Once the inputs for Clear Creek were assembled and parameter values estimated for the sites, the model's sensitivity to key parameters and inputs was examined. The purposes of the sensitivity analysis were to develop an understanding of the model's behavior—especially, how robust its predictions are to parameter values—and to identify parameters useful for calibration.

4.4.1 Methods

Six parameters were identified for sensitivity analysis because they were expected from experience with similar models to have relatively strong effects on results while having relatively uncertain values, or because they are important new parameters. They are:

- *habDriftConc*, the concentration (g/cm^3) of drift food;
- *habSearchProd* (g/cm^2), which controls the availability of benthic food sometimes used by small fish;
- *mortFishAqPredMin*, which controls the risk of predation by other fish;
- *mortFishTerrPredMin*, which controls the risk of predation by terrestrial animals;

- *maxMoveParamA*, which controls the radius over which fish can select habitat each day—increasing its value makes newly emerged fry less vulnerable to involuntary outmigration because they cannot find habitat with sufficiently low velocity;
- *fishSpawnDefenseArea*, the area (cm^2) that spawners defend around redds, which affects the frequency of redd superimposition; and
- *outmigrantSuccessL1*, the juveniles salmon length (cm) at which expected success as an outmigrant is 0.1.

In addition, the model's sensitivity to the number of spawners was examined to determine the strength of density dependent processes, especially redd superimposition. Finally, the number of habitat reaches was increased from two to eight to see how the length of simulated stream affects results.

Each of the parameters and inputs was varied over a wide range and the resulting effects on key outputs of *inSALMO* were examined. The first of these outputs is outmigration timing: how did the pattern over time in how many juveniles migrate out of the modeled system change? This result was examined by plotting the time series of daily number of outmigrants for each parameter value. The model was run for three years, water years 2007-2009.

The second and third outputs examined are the total number of live outmigrants and the total number of live outmigrants with length greater than 5.0 cm. The number of outmigrants greater than 5 cm in length was chosen because these fish stayed and grew at least a small amount within the study site, indicating the value of the modeled system as rearing habitat. The total number of outmigrants was included because it is an indicator of redd incubation success and because recent research indicates that at least a small fraction of salmon migrating out of spawning areas as fry survive to adulthood in the Sacramento River system (Miller et al. 2010).

4.4.2 Results

Drift food concentration. Varying the drift food concentration over a wide range had very little effect on when juvenile salmon migrated out (Figure 4, top panel; note the logarithmic scale) or how many total juveniles were produced (Figure 4, lower left panel). However, the number of large outmigrants was quite sensitive to drift concentration, increasing four-fold over the experiment (Figure 4, lower right panel).

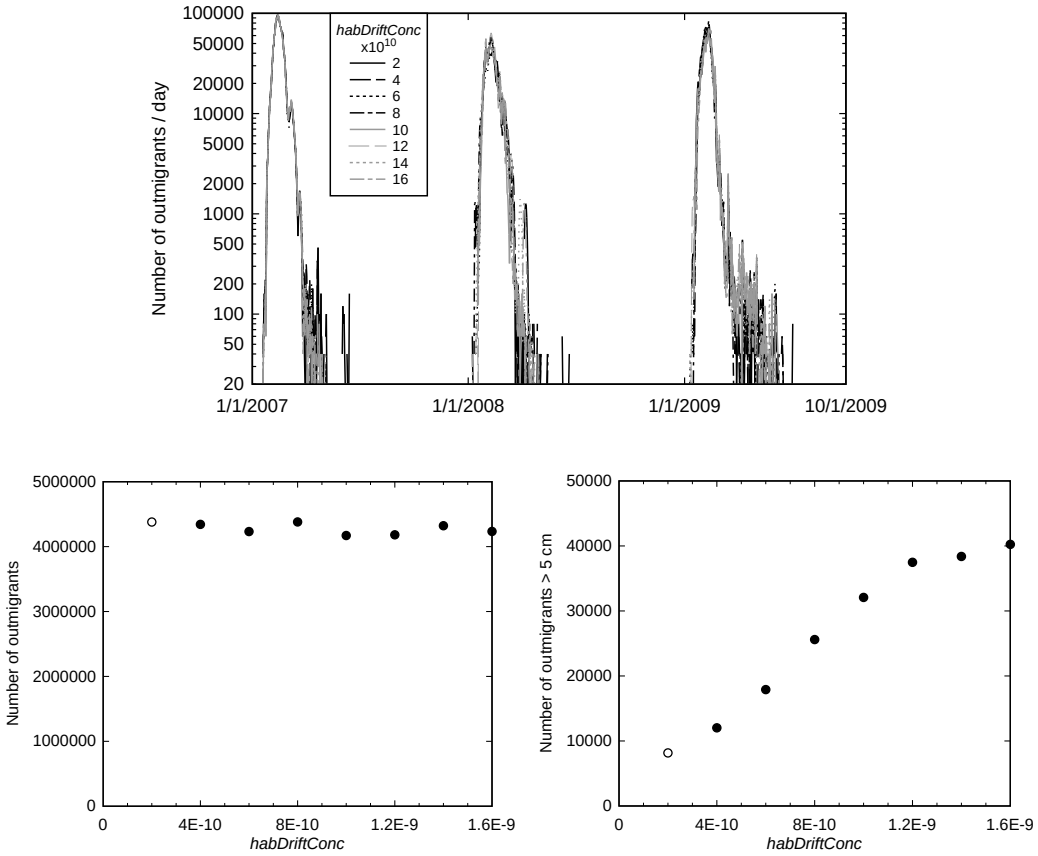


Figure 4. Sensitivity analysis results for drift food availability (parameter *habDriftConc*, g drift per cm^3). Top panel: outmigrant timing, daily number of outmigrants for the three-years run, with separate lines for the eight values simulated. Lower panels: total number of live outmigrants (left) and outmigrants greater than 5 cm length (right) vs. drift availability.

Search food production. Model sensitivity to the availability of search food was similar to that of drift food (Figure 5). Search food production had negligible effect on the timing or number of total outmigrants, but did affect the number of large outmigrants. However, search food had less effect than drift food.

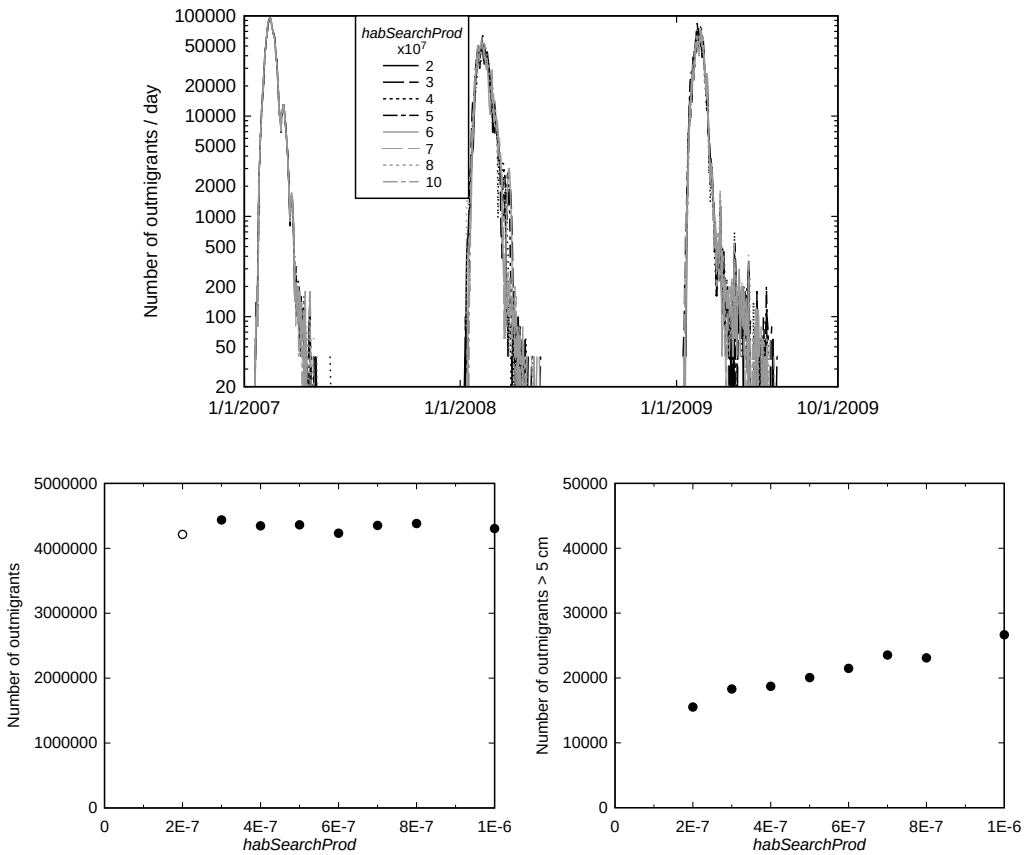


Figure 5. Sensitivity analysis results for drift food availability (parameter *habSearchProd*, g search food per day per cm^2). Format similar to Figure 4.

Fish predation risk. The probability of surviving predation by fish had very little effect on when or how many outmigrants were produced (Figure 6). The lack of effect on total number of outmigrants is no doubt because the vast majority of the juvenile salmon migrated out within a few days of emergence and hence had little time exposed to predation risk.

This parameter also had surprisingly small and complex effects on the number of large outmigrants (lower right panel of Figure 6). As survival probability increased from very low levels (0.9 to 0.94), the number of large outmigrants increased. At higher survival levels, though, the number of large outmigrants decreased. This result was in part because terrestrial predation mortality increased when fish predation decreased, a relation indicating that feeding habitat may strongly limit juvenile success. (If predation was the primary limitation, then reducing fish predation risk would let model fish select deeper habitat safer from terrestrial predators, reducing both kinds of predation.)

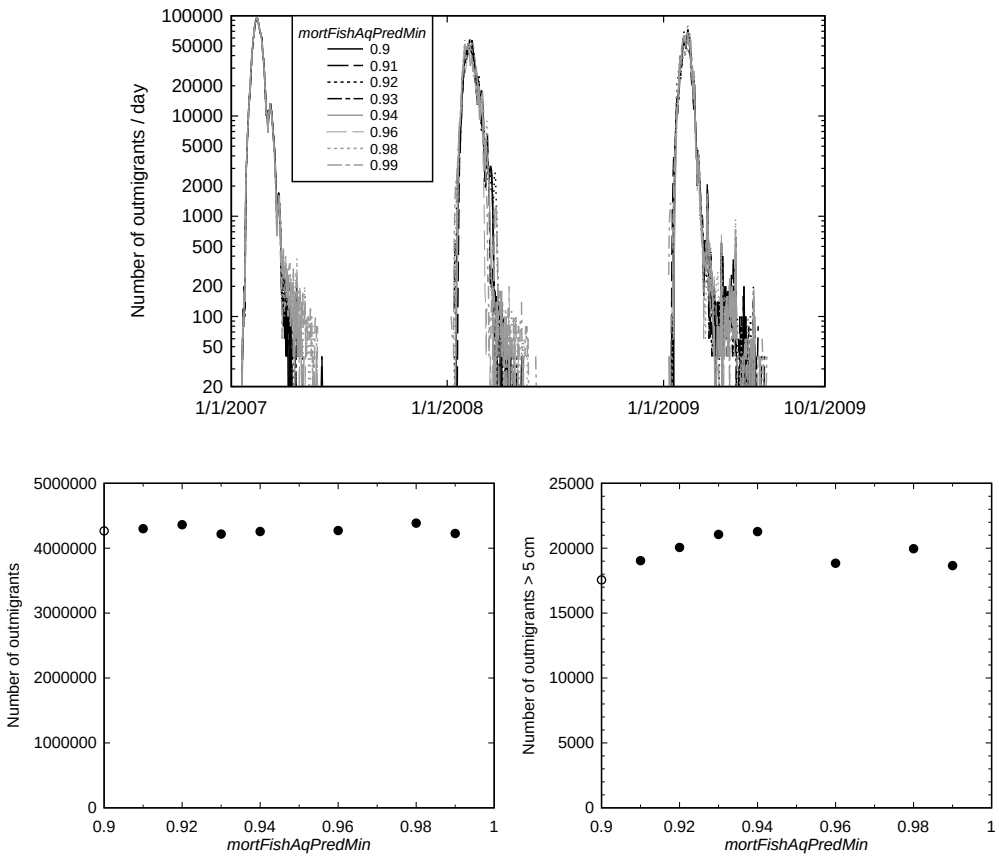


Figure 6. Sensitivity analysis results for fish predation risk (parameter *mortFishAqPredMin*; higher values represent lower risk). Format similar to Figure 4.

Terrestrial predation risk. The risk of predation by terrestrial animals also had little effect on the timing or total number of outmigrants (Figure 7), but more effect than other parameters. When predation survival was higher, a few more outmigrants stayed in the model longer, and there was a slightly positive relation between survival and total number of live outmigrants. The number of large outmigrants was sensitive to terrestrial predation at low survival levels, but not at intermediate and high levels (lower right panel, Figure 7).

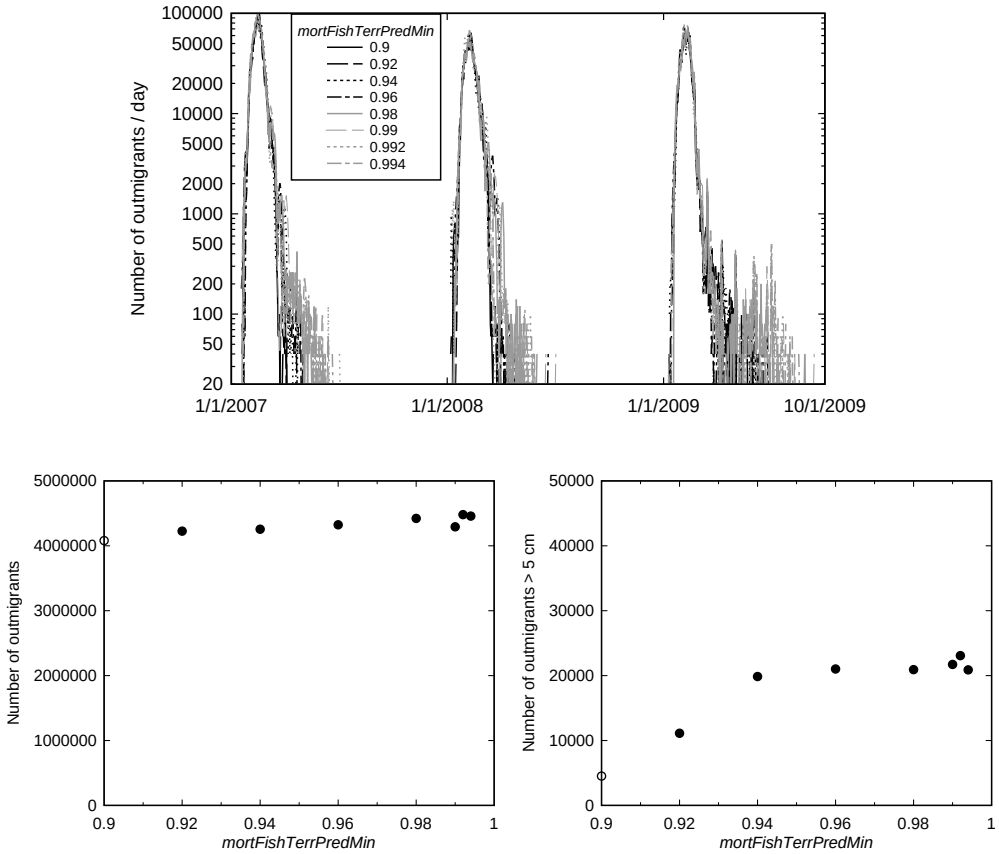


Figure 7. Sensitivity analysis results for terrestrial predation risk (parameter *mortFish-TerrPredMin*; higher values represent lower risk). Format similar to Figure 4.

Movement distance. Increasing the distance over which juveniles can select habitat had very little effect on when the vast majority of fish migrated out, indicating that the ability to reach profitable habitat may not be the reason for most outmigration (Figure 8). The number of large outmigrants did increase with the movement distance, especially above the standard value of 50.

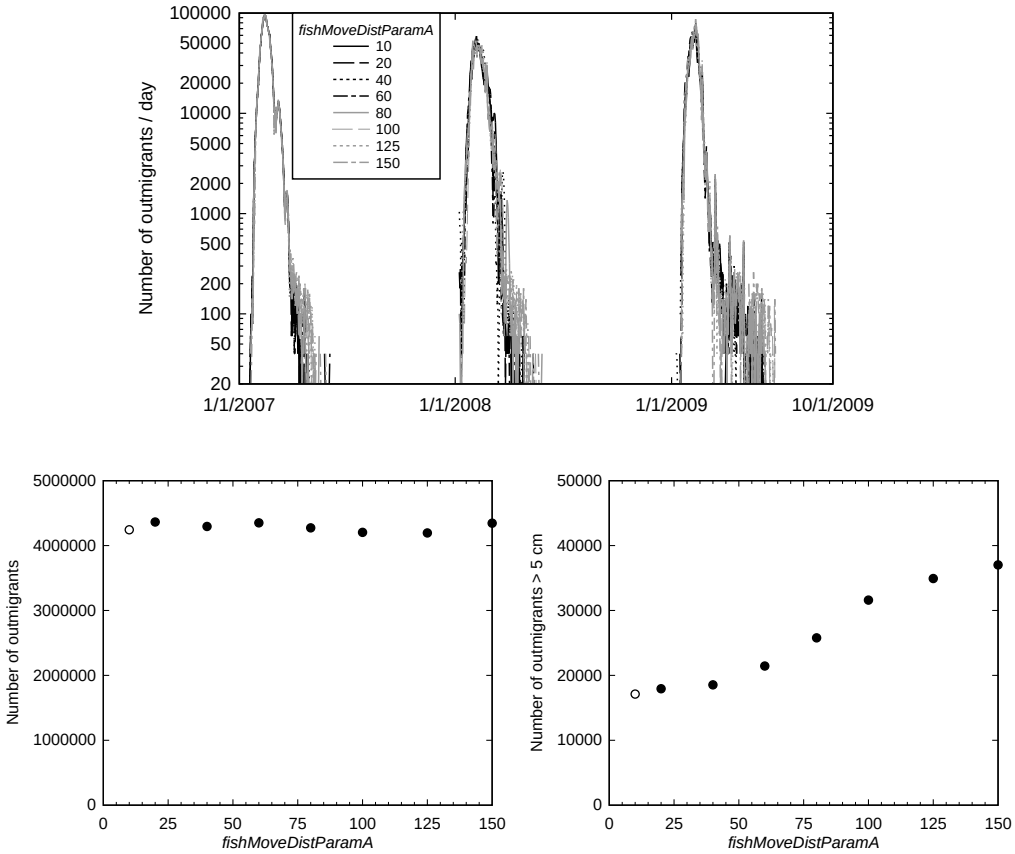


Figure 8. Sensitivity analysis results for fish movement distance (parameter *fishMoveDistParamA*; higher values let small fish select habitat over larger distances). Format similar to Figure 4.

Redd defense area. The area that female spawners defend around their redds was one parameter that did strongly affect the total number of outmigrants (Figure 9). The amount of superimposition mortality of eggs (not displayed here, but reflected in the number of outmigrants) was lowest at the third value of fishSpawnDefenseArea (7.5 m^2), and highest at both low and high values of defense area.

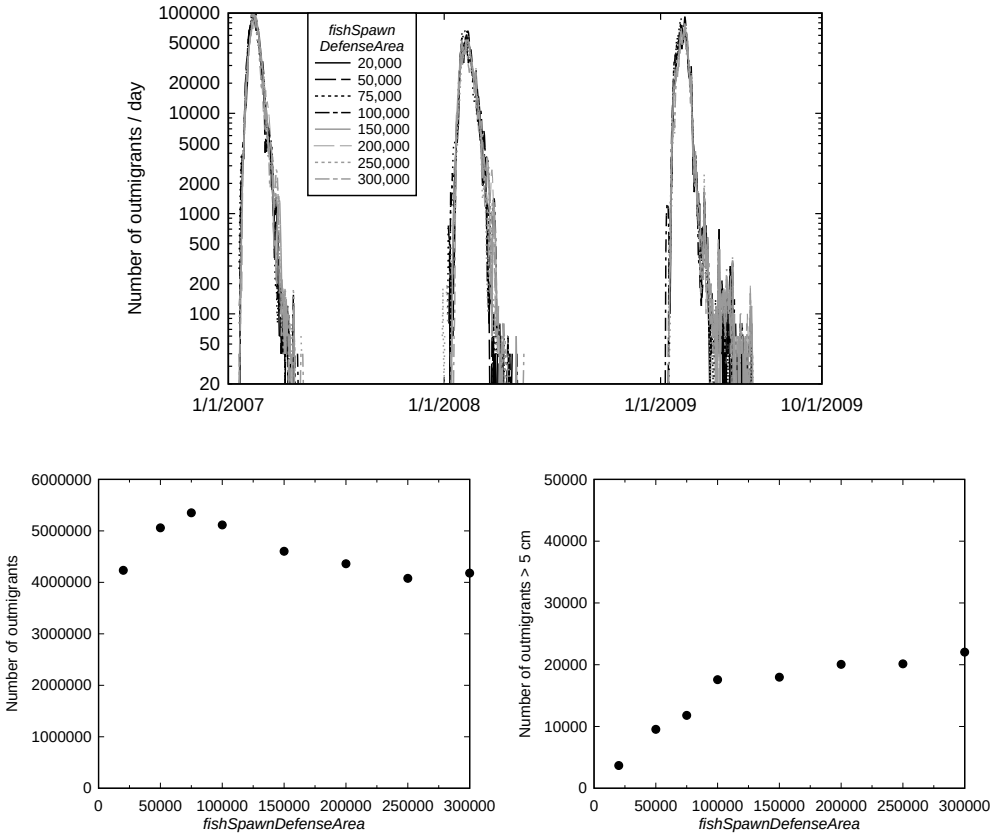


Figure 9. Sensitivity analysis results for redd defense area (parameter *fishSpawnDefenseArea*, cm^2). Format similar to Figure 4.

Outmigration success function. The shape of the logistic function of length that juvenile fish use to decide whether to migrate out had a somewhat surprising lack of effect on when all but a few juveniles migrated downstream, and (not surprisingly) little effect on the total number of outmigrants (Figure 10). The outmigration success parameter did have strong and complex effects on the number of large outmigrants. At low values of *fishOutmigrateSuccessL1* (below the standard value of 5 cm), fewer large outmigrants were produced, presumably because these parameter values encourage outmigration at smaller sizes. However, at higher than standard values the number of large outmigrants also decreased. This effect resulted from the relatively few large juveniles staying in the modeled reaches longer at higher values of *fishOutmigrateSuccessL1* (apparent from the top panel of Figure 10 and confirmed with other output), so they were exposed to predation longer.

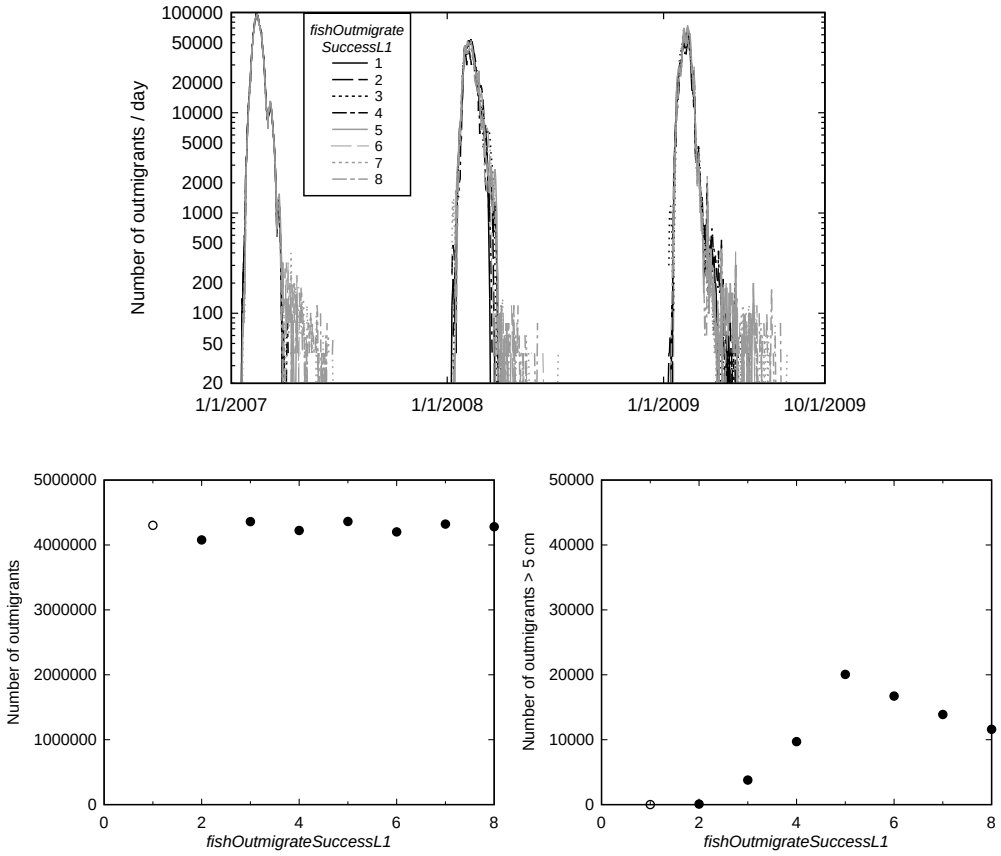


Figure 10. Sensitivity analysis results for outmigration success function (parameter *fishOutmigrateSuccessL1*, the length in cm at which fish assume relative outmigration success is 0.1). Format similar to Figure 4.

Spawner abundance. As expected, more spawners produced more outmigrants; the increase was almost linear for total outmigrants (lower left panel, Figure 11). However, the number of large outmigrants increased at a decreasing rate as the number of spawners increased, indicating density dependence. Because this density dependence occurs in the number of large outmigrants and not in the total, it appears to result from processes acting *after* emergence (e.g., competition for good feeding sites), not from superimposition. In fact, the number of

eggs lost to superimposition and the number emerging as fry both were quite linear with the number of spawners (Figure 12).

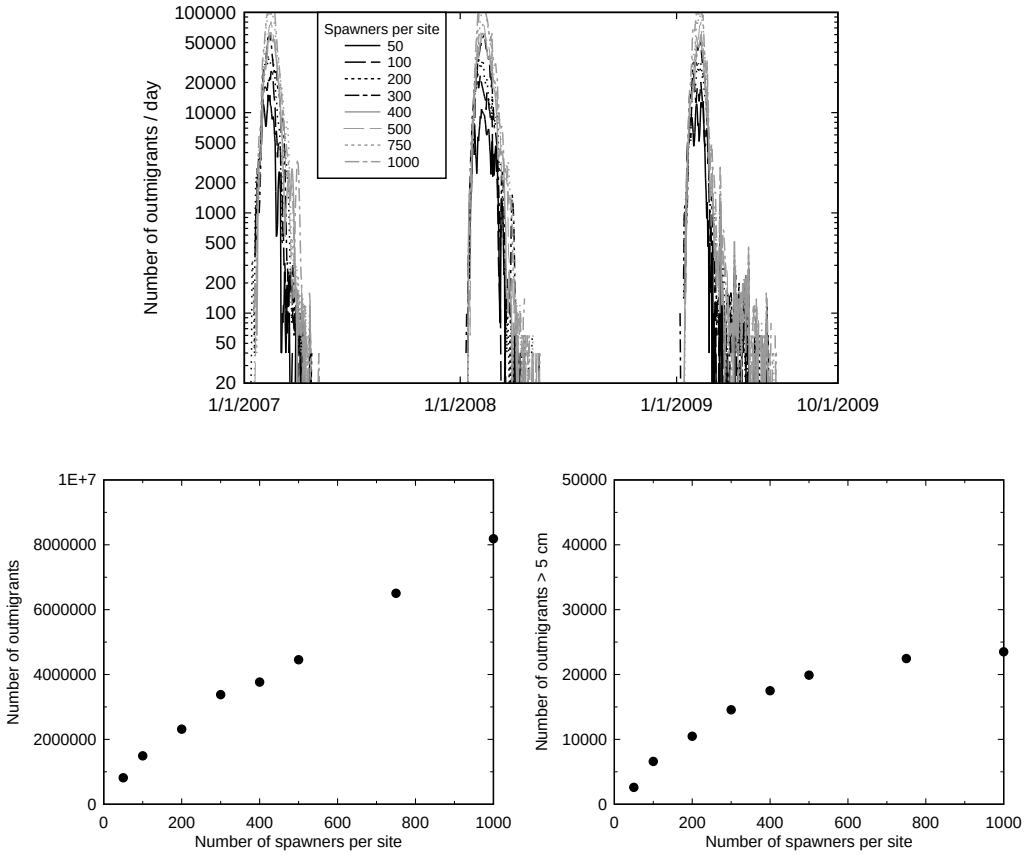


Figure 11. Sensitivity analysis results for spawner abundance (number of spawners, of both sexes, at each of the two sites). Format similar to Figure 4.

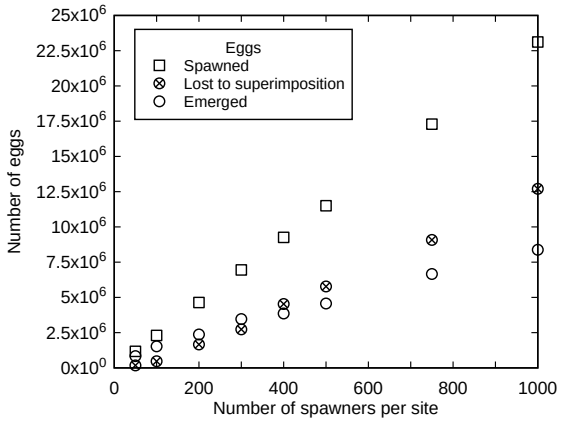


Figure 12. Response of egg survival and superimposition mortality to spawner abundance. The symbols represent the total number of eggs spawned, the number killed via superimposition, and the number surviving to emergence. Other sources of egg mortality were minor.

Number of habitat reaches. The primary question addressed by this experiment was whether traversing a longer simulated length of river would let more outmigrants grow larger before leaving the downstream-most reach. The number of reaches was increased by including copies of the two sites in the model. The four-reach scenario, for example, represents two copies each of sites 3A and 3C in an alternating sequence (from upstream to downstream: 3A, 3C, 3A, 3C). Other scenarios included six and eight reaches. The number of spawners per reach was held constant. For comparability among scenarios, results are analyzed as outmigrants per reach: the total number of outmigrants is divided by the number of reaches in the scenario. (Hence, the outmigrant results are around half those of previous experiments.)

The number of reaches had no noticeable effect on when juveniles migrated out of the downstream-most reach, or on the total number of outmigrants (Figure 13). However, simulating a longer length of stream did result in higher production of large outmigrants. When eight instead of two reaches were represented, the number of migrants with length greater than five cm increased by over 30%. Most of this increase occurred when only two more reaches were added to the model.

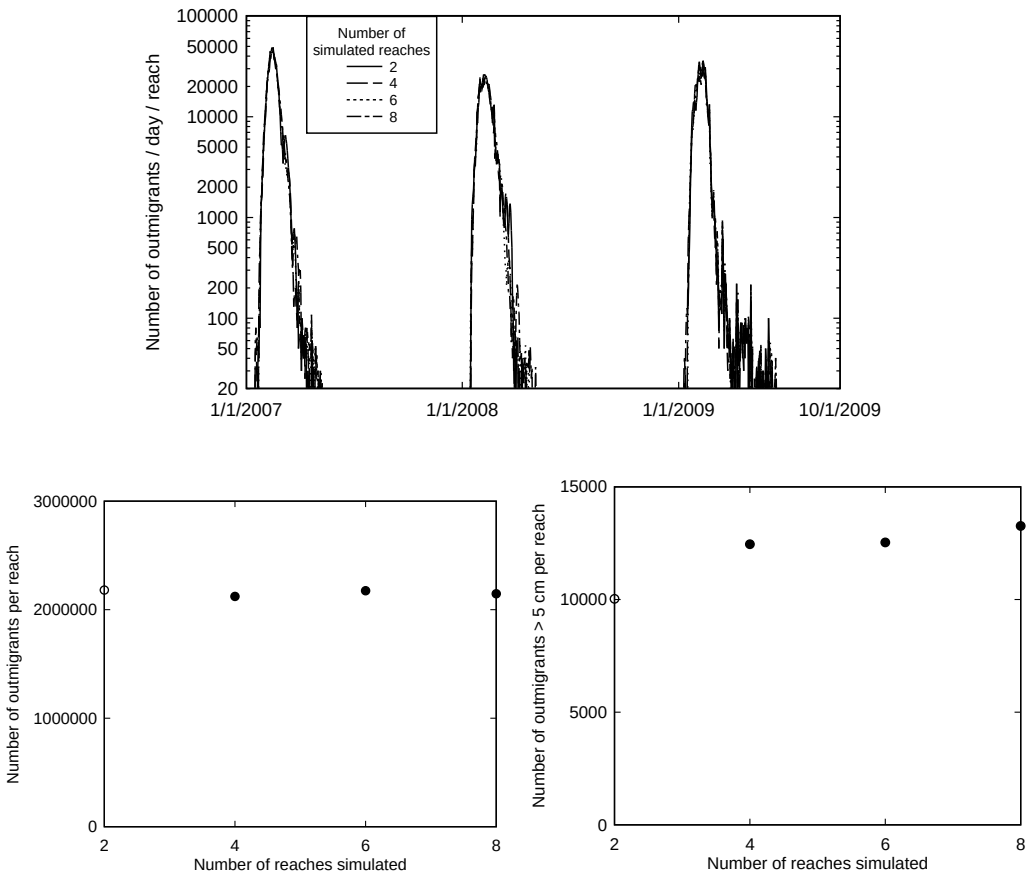


Figure 13. Sensitivity analysis results for number of reaches. Format similar to Figure 4 except that y axis is the number of outmigrants divided by the number of reaches in each scenario.

4.4.3 Sensitivity analysis conclusions

The sensitivity analysis was very valuable for understanding how *inSALMO* represents the Clear Creek study site. Several clear conclusions can be drawn.

First, none of the parameters analyzed strongly affected when the vast majority of juvenile salmon migrated downstream. The tendency for most fry to migrate out of the model almost immediately after emergence is very robust.

Parameters except those affecting superimposition had little effect on the total number of outmigrants. While parameters controlling food availability and predation risk tend to strongly affect simulations of resident salmonids, the lack of effect observed here is not surprising. Because most juveniles migrate out of the system very quickly, they are exposed to predation for very short times and have little opportunity to feed and grow.

While little affected the total number of outmigrants, the number of outmigrants that establish, feed, and survive until at least 5 cm length was sensitive to many parameters. However, predation parameters had little effect except at very low survival (high risk) values.

The number of large juveniles was density dependent, while the total number was much less so. Therefore, competition after emergence instead of superimposition seems the ultimate

limit on how many larger juveniles are produced. This result tentatively suggests that further habitat enhancement could be focused on rearing habitat more than on spawning.

Finally, the percentage of juveniles that grew before migrating downstream out of the model increased moderately with the length of the simulated stream. Doubling the stream length (using four reaches) increased the number of outmigrants longer than 5 cm by 24% and quadrupling it increased this number by 32%. This result indicates that model results are likely to underestimate the percentage of outmigrants that are large unless more reaches are represented.

4.5 Calibration and validation

Calibration is a process of adjusting particularly uncertain yet important model parameters to make the model's primary outputs better match observations. The purposes of calibration are to estimate values for the calibrated parameters while forcing the model to reproduce observations. Validation refers here simply to comparing model results to data to determine how accurately the model reproduces key patterns.

The approach taken here to calibration and validation is to compare model results to the available field data and, if it appears necessary, adjust model parameters to reproduce key patterns in the data. Several different analyses were conducted to test and calibrate different model outputs: redd locations, outmigrant timing, and outmigrant size. The methods and results of each analysis is described in a separate subsection.

4.5.1 Redd locations

The USFWS has mapped locations of Chinook redds at the study sites annually. Redd location data were provided for four years (2007-2010) that are reasonably represented by the model's input: they are after the restoration work and large injections of spawning gravel at Site 3A and site-specific temperature input are available. (These redd data are identified by USFWS via the calendar year in which they were collected; because they were observed in December, that year is one less than the water year.)

Simulated redd locations were compared to observed locations by simply plotting both on drawings of the sites' habitat cells. These displays allow qualitative assessment of how well the model's spawning site methods reproduce where salmon actually placed redds.

Table 3. Observed and simulated redd locations for December of calendar years 2007 (top) through 2009 (bottom), at site 3A.

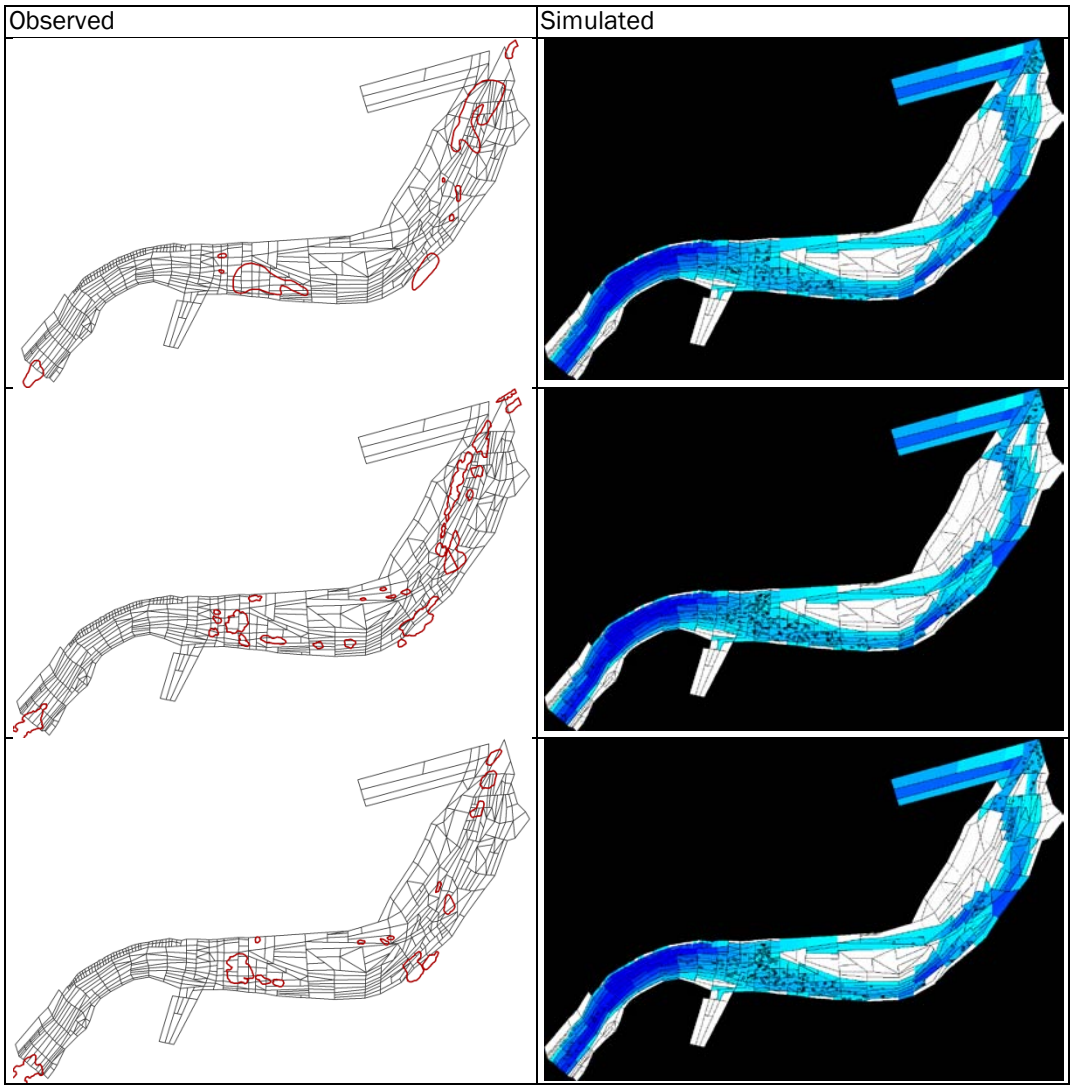
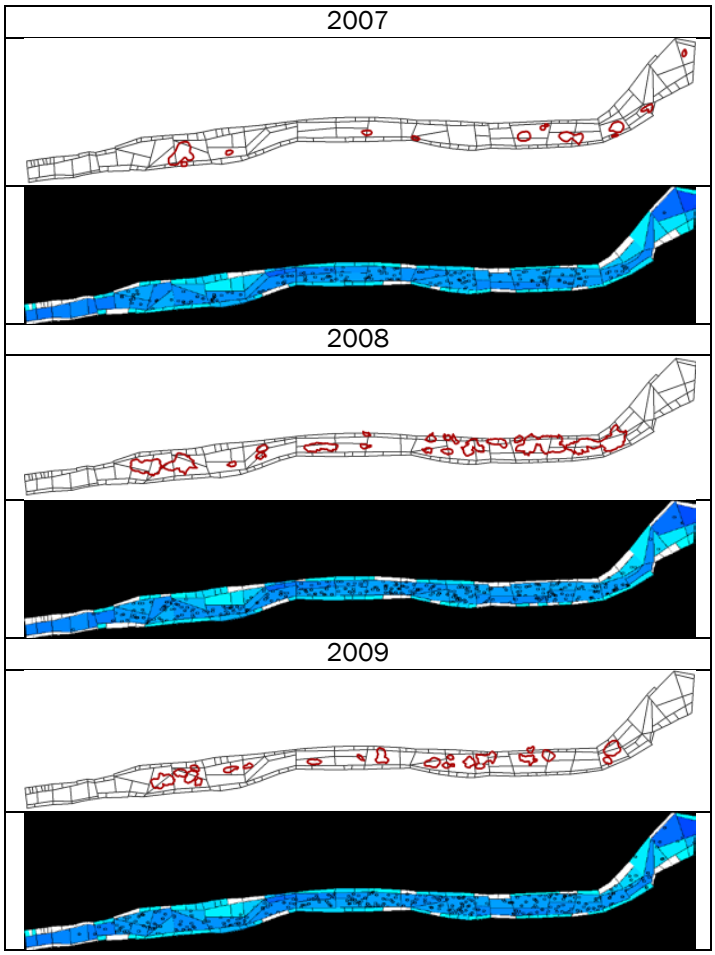


Table 4. Observed and simulated redd locations for December of calendar years 2007 through 2009, at site 3A.



This comparison of redd locations indicates that the virtual adults in *inSALMO* typically place redds in the same general locations as the real salmon. The areas that real fish clearly avoid (e.g., the large bend at the west end of 3A, both ends of 3C) are also avoided in the model. The areas of high model redd density, especially at 3A, are similar to the large areas of observed redds.

The simulated redds at 3C are more widely distributed than the observed redds. This difference could be reduced by adjusting the parameters for redd defense area or adult mortality (so spawners die sooner and hence defend redds for less time). However, the difference could also result from errors in spawning gravel input or changes over time in gravel extent, factors such as intragravel flow that affect spawning habitat quality but are not in the model, or error in the model's input for how many spawners use 3C. Given that the cause of the differences between simulated and observed redd density at 3C is not clear, and that the redd defense area parameter is relatively well supported by literature, adjusting it to calibrate redd density does not seem appropriate.

4.5.2 Outmigration timing

The main dataset available for calibration of *inSALMO* to the Clear Creek sites are the screw trap data collected by USFWS (e.g., Earley et al. 2010). The screw trap data have important limitations for this purpose. First, they capture outmigrating fish from the entire creek, not just

from the two model sites located in the downstream half of the roughly 6.75-km spawning reach used by the fall run. Therefore, the screw trap data cannot be used to distinguish between the study sites. Additionally, it is not clear how patterns in outmigrant timing and size might differ for fish spawned up to 4 km upstream of the model sites. Second, the different races or runs of Chinook salmon cannot be clearly distinguished in the screw trap data making it difficult to discern a beginning or end of fall-run outmigration.

The model's outmigration timing was compared to the screw trap data by overlaying graphs of weekly numbers of outmigrants in the (a) screw traps and (b) model results (Figure 14). Outmigration from the model reaches begins later and ends earlier than outmigration at the screw trap. Model outmigration begins in early to mid-January while the screw trap data reports early December as the beginning of fall Chinook outmigration. Model outmigration ceases between mid-April and late-June, while in some years, the screw trap continues to catch small numbers of Chinook identified as fall-run as late as September. Peaks in outmigration correspond well, typically occurring in mid-February for both the models and the screw trap.

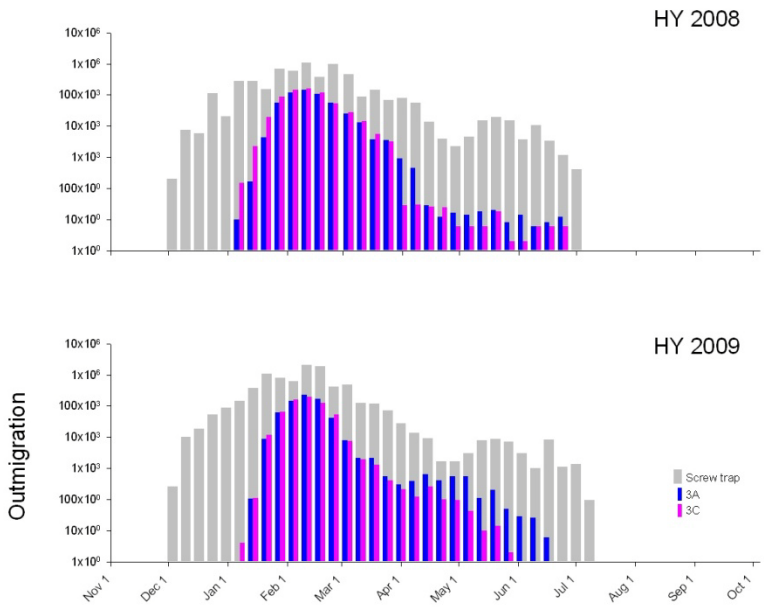


Figure 14. Outmigration timing results for water years 2008-9.

4.5.3 Outmigrant sizes

The size of outmigrants was compared to the screw trap data by overlaying graphs of weekly mean length of outmigrants in the (a) screw traps and (b) model results. Outmigrant size was calibrated to the screw trap data by varying *outmigrantSuccessL1* (the juvenile salmon length (cm) at which expected success as an outmigrant is 0.1) and visually determining which value produced a plot closest to the screw trap data for four, randomly-selected years. Setting *outmigrantSuccessL1* = 5 cm yielded the best matches.

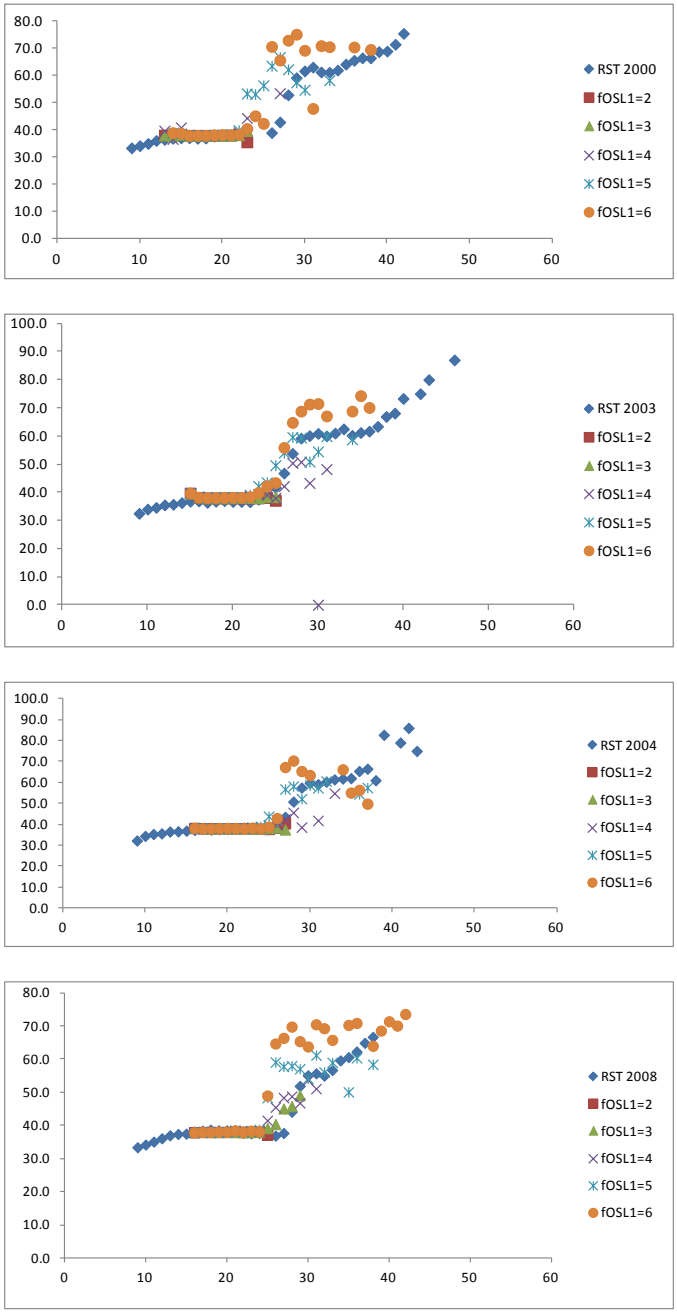


Figure 15. Outmigrant size calibration and validation results. The x axis is the water year week (week 1 starts October 1; week 10 starts December 3). The y axis is mean length (mm) of outmigrating salmon. For four example years (2000, 2003, 2004, 2008), rotary screw trap data are compared to simulation results with four values of *outmigrantSuccessL1*.

4.5.4 Total outmigrant numbers and contributions by size-class

Reflecting the spawner initialization numbers (Section 4.3.1), total annual outmigration from each of the two model reaches was compared to 7% of the annual outmigration estimated at the screw trap. For the eight years with total annual passage estimates for the screw trap, the model estimate consistently exceeds the screw trap estimate (Table 5). Attempts at calibra-

tion by increasing fish predation (reducing parameter *mortFishAqPredMin*) produce little effect on total numbers, though fish migrated out earlier.

While *inSALMO* may over-estimate total numbers of outmigrants, test runs suggest it underestimates the number of outmigrants > 5 cm FL (Table 5). For these comparisons, the model sites are treated separately because of their differences in the amount and quality of rearing habitat. Such differences are suggested by the higher proportion of larger fish outmigrating from 3A compared to 3C. Attempts at calibration through changes in *mortFishAqPredMin* and food availability (parameter *habDriftConc*) produced only minor changes to the proportion of larger outmigrants. The sensitivity analysis found that results for the number of large outmigrants can be biased by the amount of stream simulated (Section 4.4.3); including more Clear Creek sites in the model likely would improve its fit to screw trap data for this result. (The use of superindividuals also has a small effect on the simulated abundance and size of juveniles who remain and grow longer before outmigration; Section 5.3.3 of Appendix A.)

Table 5. Total outmigrant numbers and percent contribution by outmigrants greater than 5 cm fork length for the USFWS screw trap and Clear Creek sites 3A and 3C. Each of these sites constitutes 7% of the entire spawning habitat, so simulation results (columns 3 and 4 for total outmigrants; columns 6 and 7 for large outmigrants) are compared to 7% of screw trap estimates (columns 2 and 5).

Hydrologic Year	7% of screw trap (ST) total	3A total	3C total	7% ST > 5 cm	3A > 5 cm	3C > 5 cm
1999	510,000	590,000	680,000	205,000	8,000	700
2000	490,000	1,020,000	980,000	113,000	10,000	300
2002	430,000	1,240,000	1,140,000	77,000	20,000	1,500
2003	270,000	1,720,000	1,390,000	22,000	19,000	1,100
2004	420,000	1,120,000	1,040,000	17,000	9,000	600
2007	350,000	940,000	930,000	31,000	18,000	1,300
2008	390,000	540,000	640,000	47,000	8,000	1,000
2009	590,000	670,000	630,000	47,000	11,000	400

5 Example Decision-support Applications

This section documents Task 4.2 of this project, an example decision-support analysis. This analysis was designed to address typical decisions faced by water and fisheries managers in California's Central Valley. Discussions with agency staff involved in Clear Creek management identified several kinds of decisions that the model could address. Two of these were particularly important in the near-term:

- Alternative instream flow releases from Whiskeytown reservoir. Releases from Whiskeytown are the primary source of flow at the Clear Creek sites except during high runoff events. Changes in flow releases also affect stream temperatures in lower Clear Creek, so can be thought of as alternative flow and temperature regimes.
- Habitat and channel restoration projects. Several large and expensive projects to improve habitat for spawning and juvenile rearing have been undertaken in Clear Creek. Management decisions could include whether additional restoration is justified, and whether particular restoration approaches (e.g., spawning gravel augmentation; channel reconfiguration) offer greater benefits.

Each of these kinds of decisions are addressed in the example application of *inSALMO*. Section 5.1 examines the effects of habitat restoration by comparing salmon reproductive success in the 3A vs. 3C study sites. Section 5.2 illustrates use of the model to evaluate flow release management by examining the effects of instream flow changes implemented in the 1990s.

We also investigated (Section 5.3) climate change scenarios developed by the Bureau of Reclamation as part of their recent Operations Criteria and Plan (OCAP) operations review. These scenarios consisted of Clear Creek water temperature regimes for baseline and climate change conditions. We simulated the effects of the difference in temperature regime between the two scenarios.

5.1 Evaluation of habitat restoration

5.1.1 Purpose and methods

One of the most important and difficult decisions for salmon managers is whether and how to undertake extensive "restoration" projects that essentially re-build a channel to improve its productivity. These projects are very expensive, can have negative effects on other resources such as riparian habitat, and (unlike changes in flow releases) are irreversible. Another reason decisions about restoration projects are difficult is that their effects are complex and difficult to predict; even before-and-after studies of completed projects are highly uncertain because it is hard to distinguish what happens within the restored section from what happens outside it, and to distinguish project effects from other causes of variability. *inSALMO* is particularly useful for this problem; while it does not represent all the potential effects of restoration projects, it does represent the main effects such as hydraulic conditions and cover availability. Controlled experiments can be conducted in the model to distinguish the effects of restoration from other variability.

The analysis was conducted by carefully examining the differences in spawning, incubation, and rearing results between the two study sites. The two sites were chosen because 3A represents a major restoration project while 3C is an unrestored, highly disturbed site where restoration is still being considered. The simulations reported earlier (e.g., the sensitivity analyses in Section 4.4) all indicate that production of relatively large outmigrants is much higher at 3A than 3C; this analysis is to determine why that result occurs.

The model results analyzed are from five water years: 2004-08. The standard input for number of spawners at each site was used; these estimates assume a constant number of spawners per unit stream length. Because site 3A is 490 m long and 3C is 460 m, there are on average 1.07 times as many spawners at 3A. The analysis looked at differences through the stream life stages: number of eggs produced, egg survival, and the fate of juveniles after emergence. When differences between sites were found, habitat conditions and use were examined to identify explanations.

5.1.2 Results

At both sites and in all years, almost all (>94%) of females produced redds. The percent of eggs that survived to emerge as fry was relatively constant among years and sites, with the exception of unusually high survival at 3C in 2008 (Figure 16). The fate of eggs was very similar between the two sites: approximately 52% lost to superimposition and 44% surviving to emergence, with other mortality causes unimportant (Figure 17). Hence, *inSALMO* does not predict important differences between the two sites in spawning and incubation success.

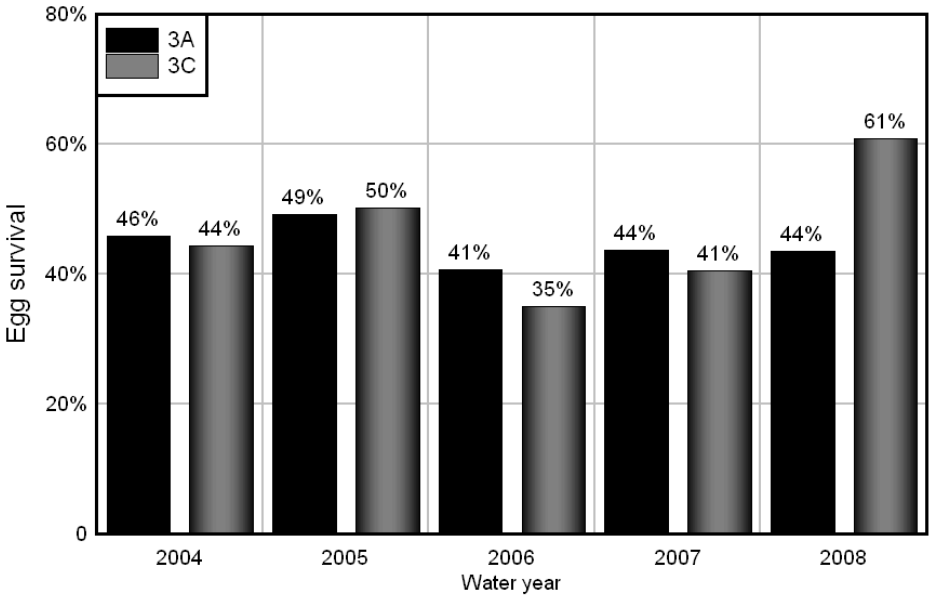


Figure 16. Egg survival (percent of eggs surviving to emerge) in the habitat restoration analysis, by year and site.

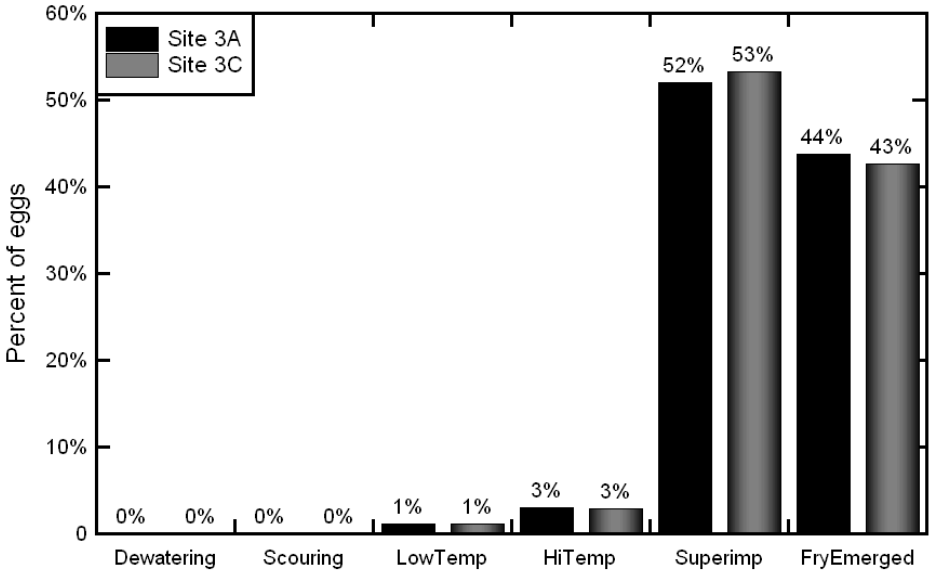


Figure 17. Fate of eggs, comparing sites 3A and 3C: percent of eggs lost to dewatering, scouring, low and high temperatures, and superimposition; and percent that emerged as fry.

At the juvenile and outmigrant stages, there were large simulated differences between the two sites. While the sites produced approximately the same total number of outmigrants per spawner, the number of outmigrants that established and grew at least to 5 cm length was much higher at 3A (Figure 18). There were no major differences in causes of mortality in juveniles between the sites (Figure 19), though higher *risk* at 3C could have caused the juveniles to avoid mortality by migrating out sooner. The number of juveniles that died at 3A was much higher than at 3C (52,300 vs. 14,500 over the entire simulation) simply because far fewer fish remained at 3C for more than 1-2 days.

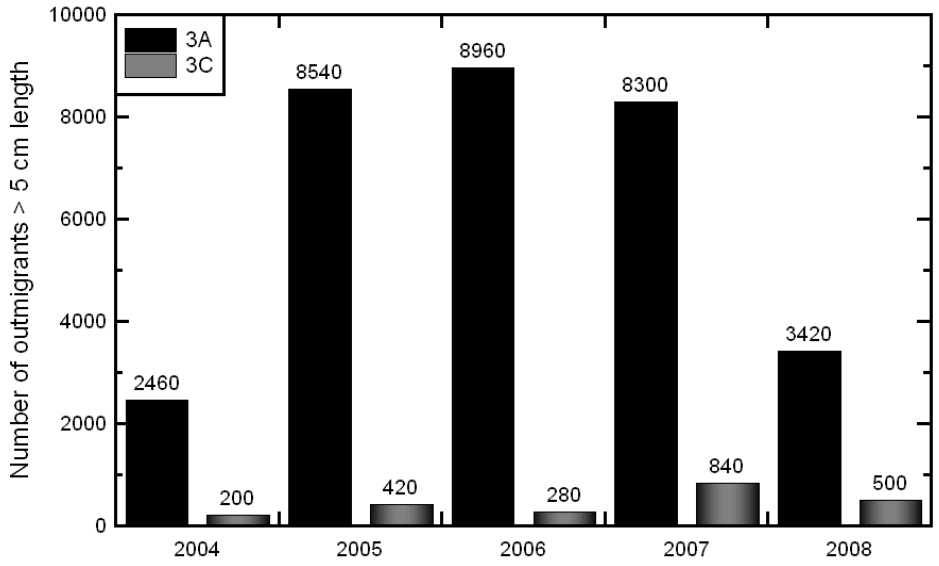


Figure 18. Number of large (length > 5 cm) outmigrants in the habitat restoration analysis, by water year and site.

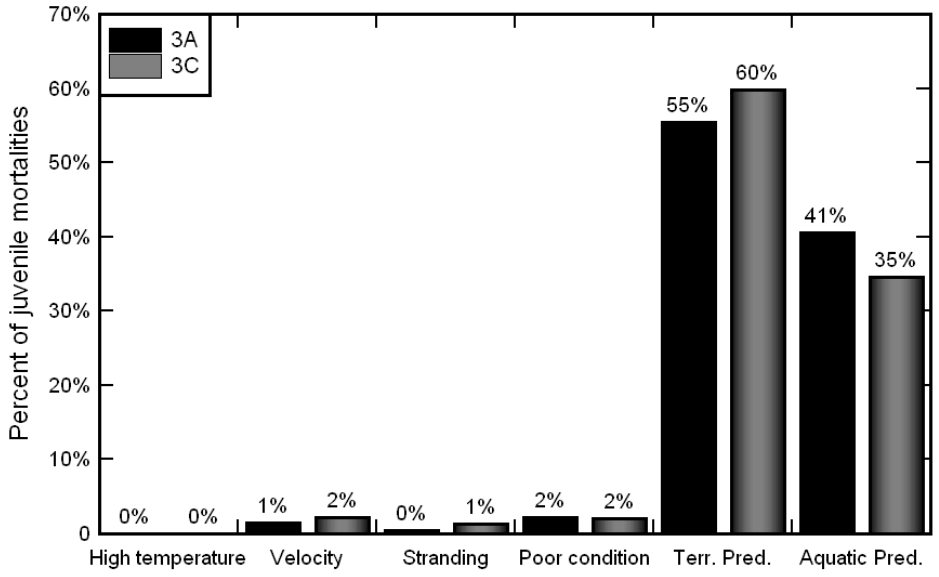


Figure 19. Causes of juvenile mortality in the habitat restoration analysis. Values are the percent of all mortalities due to each source. (The total number of fish dying as juveniles was much higher at 3A because more fish remained in the site instead of migrating out.)

Why did more emergent fry remain and grow at 3A than at 3C? This question was addressed by turning on *inSALMO*'s optional output file that reports the habitat characteristics and fish fitness variables for each fish, each day after it has selected the best available cell (the MoveReport output). This output allows examination of the habitat conditions fish were able to

find in their reach and the growth rates and survival probabilities they experience as a consequence. For example, plots showing the distribution of cell velocities and depths used by juveniles (Figure 20) indicate that a higher percentage of fish at 3A were able to find low velocities, though similar depths were used at both sites.

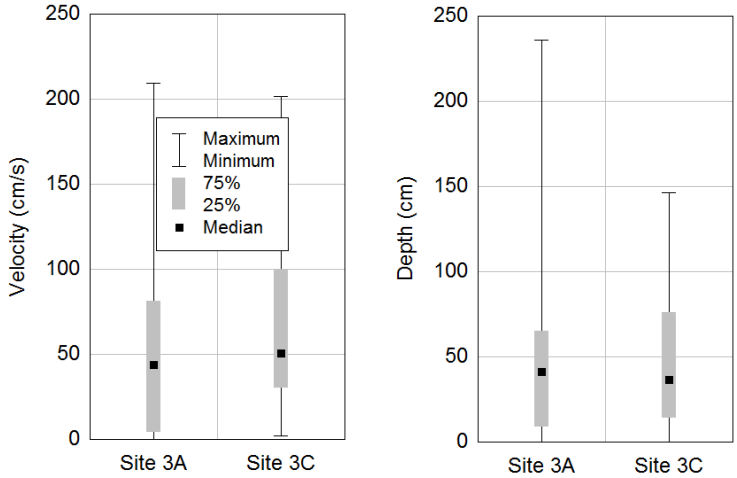


Figure 20. Distributions of cell velocity and depth for juveniles, on days when they did not decide to migrate downstream.

The differences between sites were larger for cover variables. At 3A, juveniles used velocity shelter to reduce their swimming speed 26% of the time, compared to only 6% at 3C. The median distance to hiding cover was much lower at 3A (Figure 21), which could produce a strong difference in predation risk.

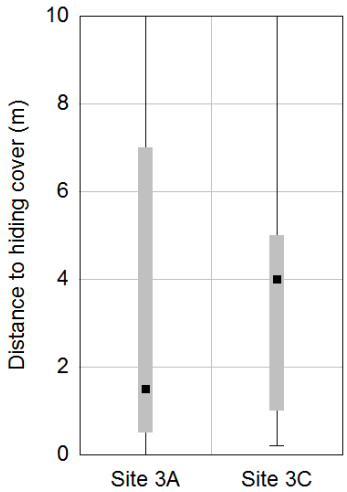


Figure 21. Distribution of cell distance to hiding cover used by juveniles, on days when they did not decide to migrate downstream.

These habitat differences did in fact produce substantial differences between sites in juvenile growth and survival potential (Figure 22). At site 3C, fewer than 25% of fish occupied cells with the positive net energy intake needed to avoid starvation and grow. At 3A, this number was above 25 but less than 50%. At both sites, many fish had extremely low survival probabil-

ities because they were unable to occupy cells with velocities below their maximum sustainable swimming speeds, but the median survival probability was much higher at 3A than 3C. The difference in median daily survival probability (0.94 at 3A, 0.85 at 3C; right panel of Figure 22) is enough to reduce the probability of surviving for 10 days from 55% to 19%.

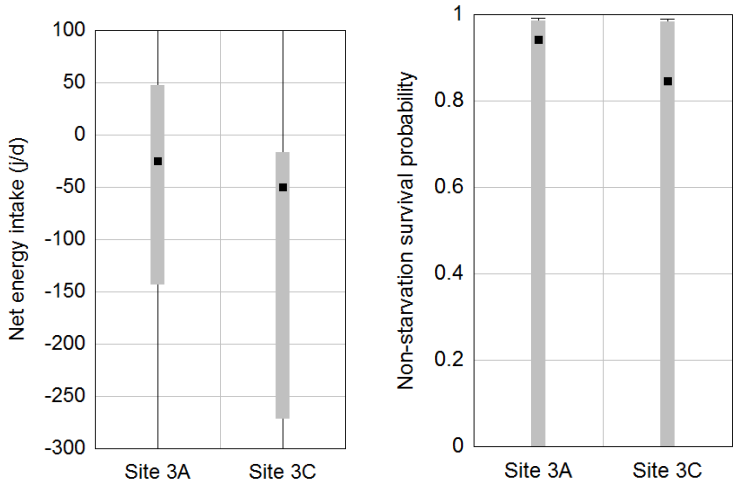


Figure 22. Differences between sites in juvenile fitness variables. Left: daily net energy intake, the difference between energy intake from food and expenditure on metabolism and swimming; fish only grow if this variable remains above zero. Right: daily probability of surviving mortality sources other than starvation. Extremely low survival probability values (e.g., < 0.8) typically result from occupying cells with velocity above the fish's maximum sustainable swimming speed.

5.1.3 Conclusions

This example decision-support application illustrates one of the primary intended uses of *inSALMO*, to design and evaluate habitat restoration projects. Its results could be relevant to decisions about future restoration work at Clear Creek.

The model application indicated that the restored site 3A has some potentially important advantages over the unrestored, degraded site 3C. The clearest difference is in the number of relatively large outmigrants produced: 3A is predicted by the model to produce approximately 14 times as many outmigrants with length > 5 cm. This length threshold indicates that the fry were able to establish and grow at the site, and (according to results reported by Miller et al. 2010) have a substantially higher potential to survive to adulthood.

The higher production of large outmigrants appears to result from higher availability of good foraging habitat for newly-emerged salmon: lower velocities, more abundant velocity shelter, and more hiding cover close to productive feeding sites. The two sites are very similar in predicted spawning and incubation success.

Some limitations of *inSALMO* 1.0 are relevant to this analysis. Especially, the model does not evaluate how restoration may affect the number of spawners using a site; instead, the number of spawners per site is fixed by model input. Because the input for this analysis assumed a constant number of spawners per unit stream length, it may have overestimated the spawner use of site 3C. The model could be used differently to address the effects of habitat restoration on where adults spawn: if both restored and unrestored areas were included in the same site, then the model would predict how many spawners place redds in restored vs. unrestored habitat. (Similarly, the model could be modified to allow adults to move among sites in selecting spawning sites.) Another potentially relevant limitation is that *inSALMO* does

not currently assume any effects of spawning gravel quality, so benefits of restoration to the flow of water through redds, etc., would not be reflected in results.

5.2 Evaluation of alternative instream flow and temperature regimes: Before 1995 vs. after 2001

5.2.1 Purpose and methods

Another of the primary purposes of *inSALMO* is to evaluate alternative flow and temperature scenarios. This use of the model is demonstrated here by evaluating and contrasting two historic periods that differ clearly in flow and temperature regime. Flow releases from Whiskeytown Reservoir into Clear Creek were gradually increased over the years 1995 to 2001, with resulting changes in temperature. Hence, the periods before 1995 and after 2001 are distinctly different, as illustrated in Figure 12 and Figure 23. This analysis contrasts two eight-year periods: the "before" period of water years 1986-1994 and the "after" period of water years 2002-2010. Flow in the after period averaged 4.1 m^3/s (150 cfs) higher than in the before period, and the year-round average temperature was 2.5°C lower in the "after" period.

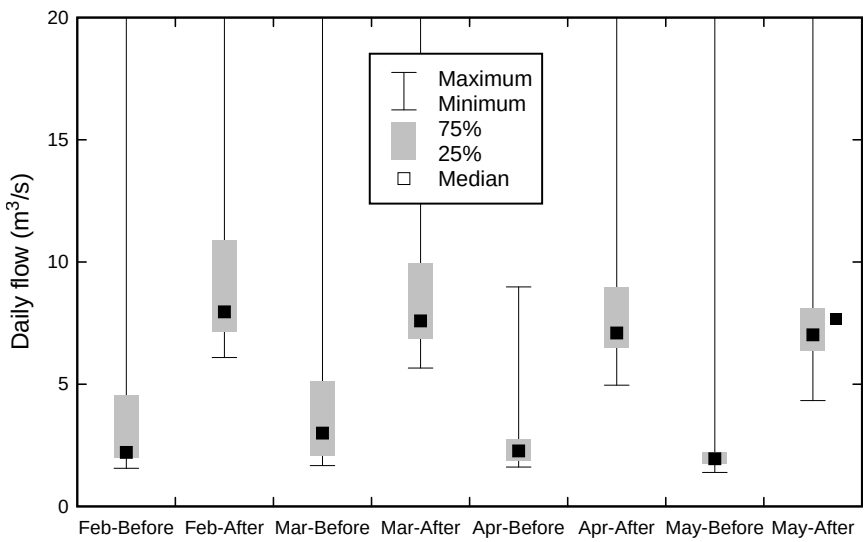


Figure 23. Differences in flow regime between (a) water years 1986-1994 ("Before") and (b) water years 2002-2010 ("After"). The box-and-whisker plots represent the distribution of daily mean flows, for months February-May, for the before and after periods.

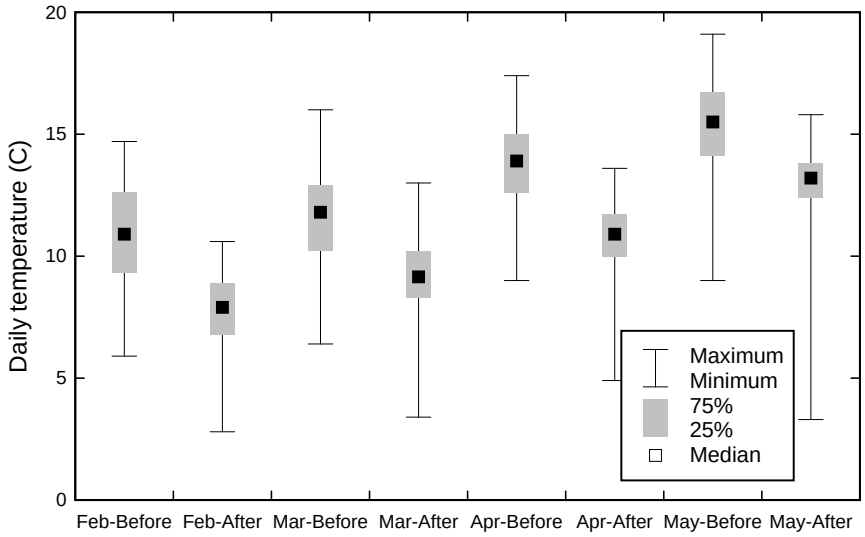


Figure 24. Differences in temperature regime between water years 1986-1994 and 2002-2010; format is the same as that of Figure 23.

The two flow and temperature regimes were contrasted by simulating their effects on salmon spawning and outmigrant production while holding other factors such as physical habitat and spawner abundance constant. Differences in results between the two regimes were investigated by examining the separate spawning, incubation, and juvenile rearing stages.

The model runs used standard input except for spawner numbers. Daily flows and temperatures were from the input described at Section 4.2.1. Spawner numbers were held constant to eliminate them as a source of difference between the "before" and "after" periods, and because spawner counts are not available for part of the "before" period. Each year, sites 3A and 3C were assumed to receive 640 and 600 spawners that were 57% female; these are approximately the averages of observed values. Five replicate simulations were executed for each of the two scenarios.

5.2.2 Results

Differences in redd numbers and egg production were examined first. Fewer redds and eggs were produced in the "before" period: the total number of redds produced per eight-year period was 4824 before (standard deviation among the 5 replicates = 32) and 5458 (SD = 13) after the flow increase, an increase of 13%. The reason for this difference appears to be that temperatures in the "before" period were high enough during the October spawning season to frequently delay spawning. The parameter for maximum temperature at which Chinook will spawn was set to 14C, often exceeded in the "before" period but not "after" (Figure 25). Hence, adults delayed spawning and some died of predation or poor condition before spawning. The average date of spawning was 12 days later before the flow change.

There was little overall change in simulated egg survival from the difference in flow regime. From "before" to "after" periods, egg mortality due to high temperatures decreased from 6% to 3% of eggs, but superimposition mortality increased from 48% to 50%. Other sources of egg mortality were negligible in all cases. Overall egg survival was 45-46% in both periods. Even though spawning occurred on average 12 days earlier in the "after" period, emergence occurred on average 4 days later. This difference is due to slower egg development at lower temperatures, and because the earliest redds (most likely to be delayed by high temperature before the flow change) tend to be destroyed by superimposition and not affect mean emergence time.

The overall effect of all these spawning and incubation processes is that 12% more fry emerged in the “after” period, on average 4 days later than before the flow change.

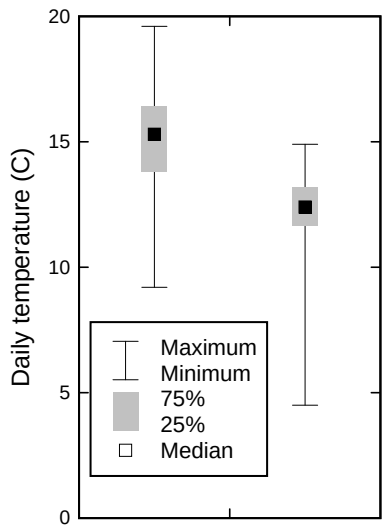


Figure 25. October temperature distributions in before and after periods.

The increased instream flow had little effect on the total number of outmigrants but strong, and negative, effects on the number of fry that established and grew to at least 5 cm length before migrating out. The total number of outmigrants was 12% higher in the “after” period, reflecting the difference in fry emerging from redds. However, the number of outmigrants with length > 5 cm was 38% lower in the “after” period.

Lower production of large outmigrants with higher flow is likely due to reduced availability of moderate-velocity habitat. The decrease in large juveniles occurred in both sites 3A and 3C, indicating that the habitat restoration in 3A did not offset the negative effects of higher flow (Figure 26). The timing of outmigration is illustrated in Figure 27. More large juveniles migrated out earlier in the “before” period.

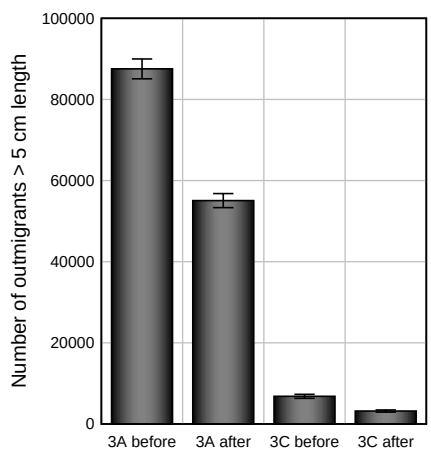


Figure 26. Number of large (length > 5 cm) outmigrants produced before and after the instream flow increase, by site. Error bars indicate 1 standard deviation over the 5 replicate simulations.

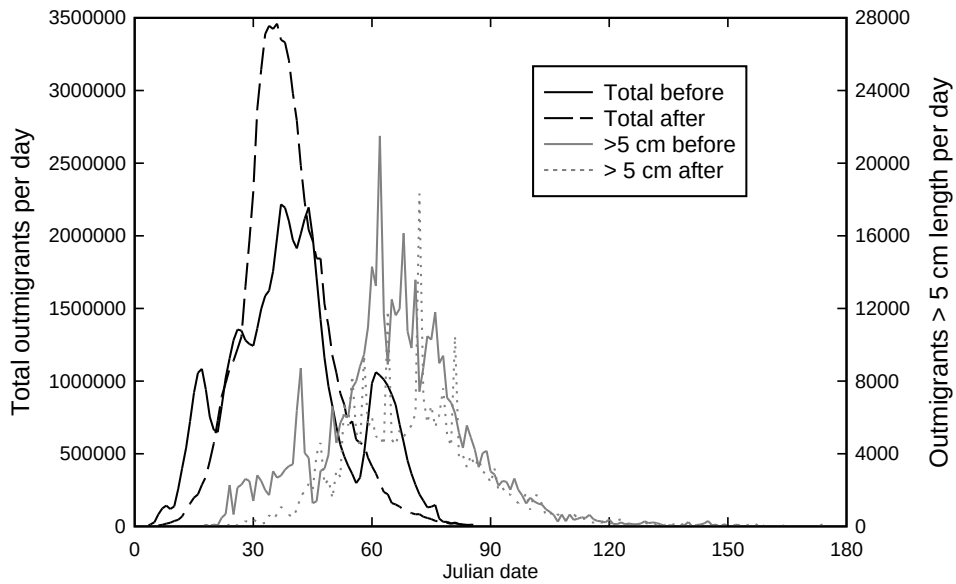


Figure 27. Outmigration timing, before and after the increase in instream flows. The X axis is the day of the calendar year (1 = January 1; 90 = March 31), and the Y values are the average number of outmigrants per day, over all replicates of all simulated years. The left Y axis refers to total outmigrants (black lines), and the right axis refers to large outmigrants (grey lines).

5.2.3 Conclusions

The *inSALMO* simulations identified a number of ways that a change in flow and temperature regime can affect spawning, incubation, and juvenile rearing. The first difference identified between the "before" and "after" periods is that higher October temperatures before the flow change delayed spawning by exceeding the maximum temperature at which *inSALMO* lets adults spawn. This process is a very simplified representation of high-temperature effects on spawning, but the general result that adults delay spawning until temperatures are low enough to avoid egg mortality is reasonable. The model predicted that adult mortality during the delay slightly reduced redd production.

The strongest effect of the change in flow regime was predicted to be on rearing and growth of newly emerged juveniles. The increase in flows actually appears, from the simulations, to reduce the availability of feeding sites that provide positive growth and safety from predation.

5.3 Comparison of OCAP climate change scenarios

5.3.1 Purpose and methods

The recent Central Valley Project "Operations Criteria and Plan" study conducted by the Bureau of Reclamation assessed biological impacts of the Project's continued long-term operations. Part of the assessment was prediction of water temperatures under several climate change scenarios. Temperatures were predicted using the Bureau's water temperature models, which predict daily stream temperatures at a number of locations, using historic conditions from water years 1922 through 2003 as a basis. The meaning of the predicted temperature changes to resources such as Chinook salmon were not predicted with the kind of detail that *inSALMO* can provide.

The OCAP climate change study developed six climate change scenarios (USBR 2008). The two most different scenarios are analyzed here. "Study 9.0" is the baseline scenario, assum-

ing historic climate conditions. The “Study 9.5” scenario assumes a drier and warmer climate with sea level rise.

The effects of these two scenarios on Clear Creek Chinook were simulated by using their water temperature values as input to *inSALMO*, while using standard values for all other inputs (including flow). The OCAP analyses produced water temperature values for sites at the Igo gage (Section 4.2.1) and at the mouth of Clear Creek. To represent the *inSALMO* reaches, the two OCAP temperatures for Igo and the mouth were averaged.

The simulations therefore used two scenarios: Study 9.0 and Study 9.5 water temperatures. Five replicates were used. Simulations ran for five water years, 1999 through 2003, the entire period of overlap between the OCAP temperature values and the *inSALMO* standard input. These two scenarios differed in temperature little in winter and spring, and by 1-2°C in summer–fall (Figure 28).

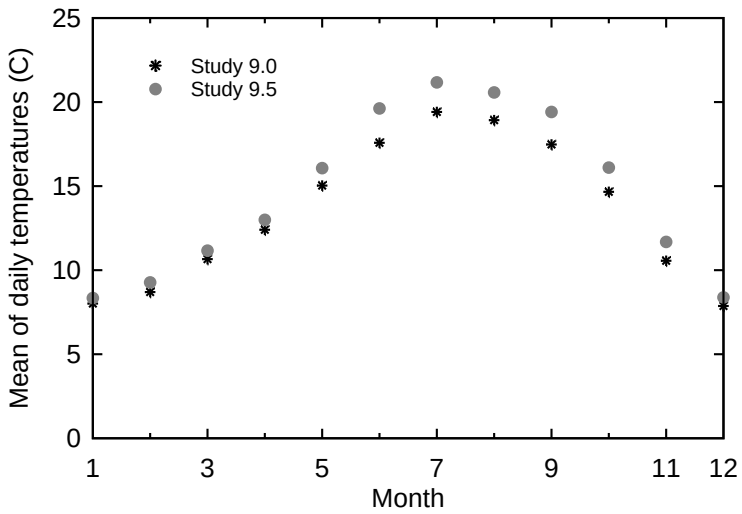


Figure 28. Temperatures representing the two climate change scenarios. Values are the mean of daily temperatures, by month, over the 1999–2003 period simulated.

5.3.2 Results

The Study 9.5 climate change temperatures were predicted by *inSALMO* to affect the fate of Chinook eggs but in ways that summed to little overall difference in the number of emergent juveniles. The average number of redds produced was 9% less under Study 9.5, compared to Study 9.0. This difference resulted from higher temperatures under Study 9.5 causing females to delay spawning, which resulted in the death of some females before they could spawn. The average date of spawning was 11 days later under Study 9.5 even though adults arrived over the same dates in both scenarios.

The reduced number of redds under Study 9.5, and a small increase in high temperature egg mortality, was almost completely offset by decreased superimposition mortality (Figure 29). The net effect was only a slight decrease in the number of successful eggs due to the increased temperatures.

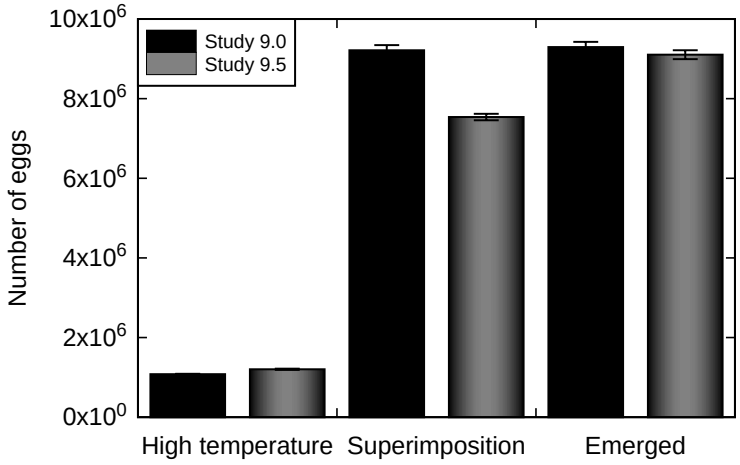


Figure 29. Egg fates in the OCAP climate change analysis. Bar heights are the mean number of eggs over five replicates, using the total over the five years simulated. Error bars represent the standard deviation over the five replicates. “High temperature” and “Superimposition” are the number of eggs dying of these causes; “Emerged” is the number of eggs surviving to emergence.

The strongest effect of the Study 9.5 climate change scenario was in the number of relatively large outmigrants produced. This scenario produced only 2% fewer total outmigrants than the baseline Study 9.0 scenario, but 16% fewer outmigrants with length greater than 5 cm. This decrease in large outmigrant production seems surprising, given the quite small temperature differences during the outmigration and rearing periods (January–May; Figure 28). Further analysis showed that this result may in part be a coincidence. The delay in spawning caused by higher fall temperatures in the Study 9.5 scenario translates into emergence being delayed by an average of seven days. As a consequence, the peak of fry emergence occurred on dates (around the first week of February) that happened to coincide with high-flow events in several of the simulated years. High flows during emergence causes the simulated juveniles to migrate downstream immediately because they offer poor survival and growth conditions.

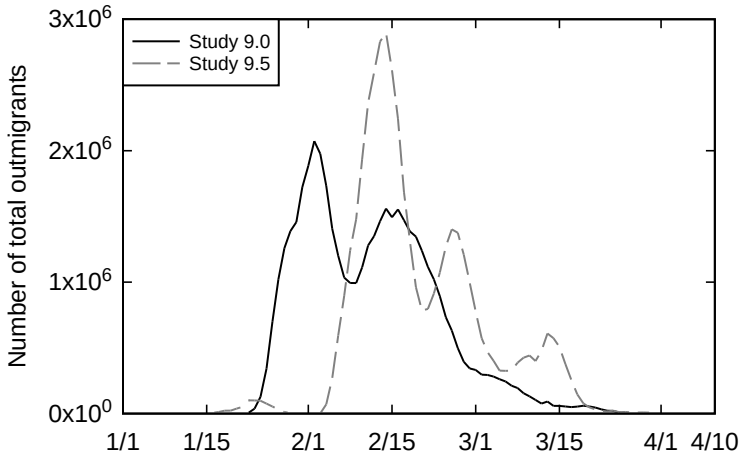


Figure 30. Outmigration dates in the climate change analysis. The x axis is the day of the year (month/day) and the y axis is the total number of outmigrants over five replicates of five-year simulations.

5.3.3 Conclusions

Overall the *inSALMO* simulations predicted that climate change, as represented by the difference between the OCAP Study 9.0 and 9.5 temperature scenarios, would have quite small effect. This is not surprising given the relatively small differences between these two scenarios (Figure 28). One effect was in delaying the start of spawning because water temperatures with climate change exceeded the *inSALMO* parameter representing the highest temperature at which female adults will spawn often at the start of the spawning period. However, the resulting decrease in redd production was largely offset by reduced superimposition mortality: redds produced early in the spawning period are often destroyed by superimposition anyway.

6 Limiting Factors Analysis for the Clear Creek Application

6.1 Objectives

Task 7 of this project is to develop tools for "limiting factors analysis": use of *inSALMO* to determine which habitat and management factors have strong effects on production of juveniles salmon. The product of Task 7 is the "Limiting Factor Tool" (LFT), an element of the *inSALMO* GUI that automates simulation experiments to evaluate the importance of 10 factors. The LFT is documented in Appendix D.

This section applies the LFT to the Clear Creek study sites. The objectives of the analysis are to illustrate use of the LFT, and to identify the factors that, according to this application of *inSALMO*, have strong effects on juvenile salmon production.

6.2 Methods: Application of the Limiting Factors Tool

The LFT was applied to the Clear Creek sites using standard input and parameter values for *inSALMO* (as described in sections 4.2 and 4.3 and Table 1 and Table 2). A three-year period, water years 2007-9, was simulated.

The LFT has its own set of parameters that control the experiments it executes. The parameter values listed in Table 1 of Appendix D were used. Seven scenarios were executed for each factor. The experiments are summarized here.

- Base flow was varied over a range of -2 to +4 m^3/s from actual values.
- Food availability was varied over a range of -50% to +200% of the standard values.
- Hiding cover availability was varied from -50% to +150% of standard input.
- Daily risk of piscivory was varied ±10% of the standard value.
- Redd depth (which affects scour risk) was varied from -50% to +150% of the standard value of 20 cm.
- Spawning gravel availability was varied from -50% to +150% of standard input.
- Summer temperature was varied from -4 to +4 °C from May through October.
- Winter temperature was from -4 to +4 °C from November through March.
- Velocity shelter availability was varied from -50% to +150% of standard input.
- The number of spawners was varied from -50% to +150% of standard input.

Uncertainty was represented by running each factor scenario for the eight combinations of low and high values of the parameters *fishRespParamA*, *fishMaxSwimParamA*, and *fishOutmigrateSuccessL1*. Hence, *inSALMO* was executed 56 times for each factor. (LFT experiments this big are not currently feasible in Windows, so the model runs were executed in Linux.)

6.3 Results

The LFT produces both statistical results and graphs showing the effect of each parameter. The statistical results summarize the strength of each factor's effect as the slope magnitude (the absolute value of slope, so strong negative effects are comparable to strong positive effects) and R^2 of a linear regression between the factor (scaled between zero and one) and the model results, with separate statistics reported for total number of outmigrants and the number of large (length > 5 cm) outmigrants. Factors with highest slope magnitude and R^2 are indicated as most "limiting" juvenile salmon production.

The statistical results for the Clear Creek application (Table 6) indicate that different factors were most important for total vs. large outmigrants. The factors with strongest effect on the total number of outmigrants were winter temperature and the number of spawners. Spawning gravel was also important; its slope was considerably less than that of winter temperature,

but its R^2 value much higher, indicating a more consistent effect (which is explored more below). For the number of large outmigrants, the most important factor by far was food availability. Winter temperature and velocity shelter were the next most important factors. Spawning gravel and number of spawners were much less important for large outmigrants than for the total number of outmigrants.

Table 6. Limiting factors analysis statistical results.

Factor	Total outmigrants		Large outmigrants	
	Slope magnitude	R^2	Slope magnitude	R^2
Base flow	40562	0.02	1869	0.01
Food	42927	0.03	19082	0.24
Hiding cover	70137	0.06	490	0.00
Piscivory risk	40199	0.02	764	0.00
Redd scour	0	0.00	0	0.00
Spawn gravel	1615699	0.79	1876	0.01
Summer temperature	23458	0.00	2846	0.01
Velocity shelter	25482	0.01	4077	0.03
Winter temperature	4108679	0.50	8267	0.11
Number of spawners	3856091	0.99	3176	0.02

While these summary statistics are useful for quick, concise comparison of factors, they conceal a great deal of information about how the model responded to each factor. To get a more complete picture of results, it is imperative to examine the LFT's graphical results: the regression plots from which the above statistics were taken.

One general result from the regression plots (Figure 31 through Figure 40) is that the total number of outmigrants varied much less than the number of large outmigrants among the scenarios and uncertainty parameter combinations. This result was also obtained in the sensitivity analysis (Section 4.4). As a consequence, R^2 values tend to be higher for the total outmigrant results (except for factors that had little effect on total outmigrants).

The summary results for base flow in Table 6 and the regression line in Figure 31 indicate that higher instream flows had a generally negative but not extremely strong effect on production of bigger outmigrants. Close inspection of the base flow results shows that reducing flow had little effect. (Remember that the experiment varied flows from -2 to +4 m^3/s from actual values, so the middle scenario represents an increase of 1 m^3/s.) However, increased flow—the middle scenario and higher—had relatively strong negative effects on the number of large outmigrants.

Food availability (Figure 32) had virtually no effect on the total number of outmigrants but a strong positive effect on big outmigrants.

Hiding cover (Figure 33) and piscivory risk (Figure 34) had little consistent effect and hence are not indicated to be important factors limiting production of juvenile salmon. Redd scour had no effect because no redd scour occurred under any of the scenarios (Figure 35). Spawning gravel availability (Figure 36) affected total but not large outmigrant production, as in the sensitivity analysis results investigated in Section 4.4.2.

The water temperature results especially illustrate the limitations of the statistical summary of Table 6. The statistics show a small positive effect of summer water temperature on the number of big outmigrants, which may seem counterintuitive. The graphical results, however, show that production of large outmigrants increased with summer temperature but not at the highest temperature scenarios (Figure 37). Hence, the model indicates that temperatures observed in 2007-9 are in a near-optimal range; substantial increases or decreases are predicted not to be beneficial and small changes are unlikely to have noticeable effects.

Winter temperature results are even more interesting and complex (Figure 38). The model predicted zero redd survival at the lowest winter temperature scenario, because of low-temperature egg mortality. This result indicates that any management change that would produce substantially lower winter temperatures should be considered very carefully. The parameters for low-temperature survival of eggs used in *inSALMO* do not have an especially firm basis (Table 2), so additional research on low-temperature egg survival could be valuable in the case that lower temperatures are considered. The winter temperature experiment also produced a highly nonlinear response: while the lowest temperatures produced no outmigrants, the highest numbers of both total and big outmigrants occurred at intermediate winter temperatures. Again, the observed 2007-9 winter temperatures appeared to be near-optimal.

The LFT results for number of spawners parallel the sensitivity analysis results: more spawners produced proportionally more total outmigrants, but the response of big outmigrants was less strong and flattened as spawner numbers increased (Figure 40).

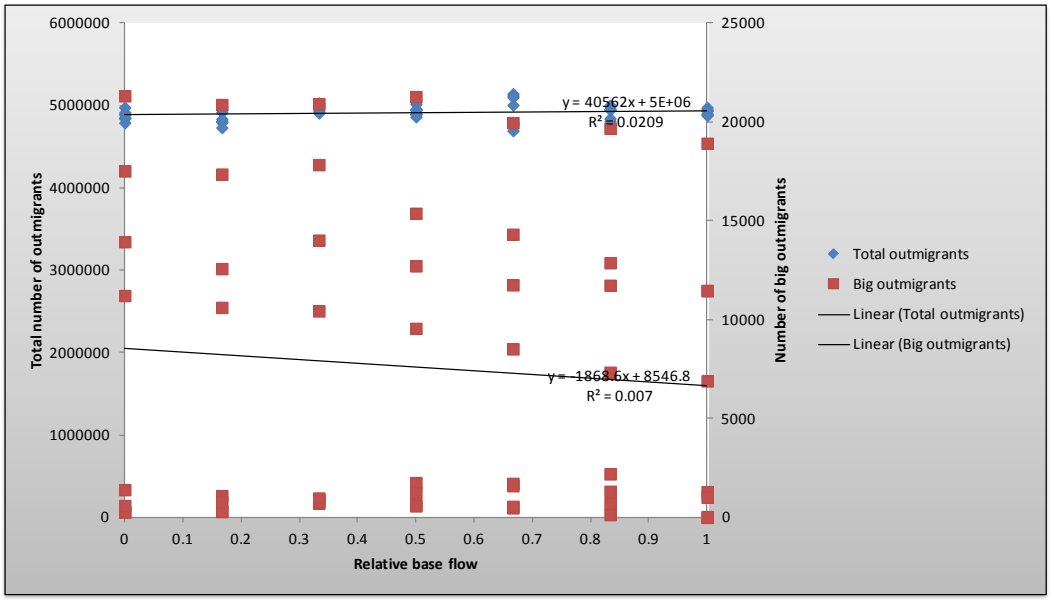


Figure 31. LFT results for base flow. The x axis is the scaled value of base flow: x = 0.0 is the lowest base flow scenario, x = 1.0 is the highest flow scenario, and intermediate x values represent the scenario's flow as a fraction of the difference between lowest and highest. The left y axis indicates the total number of outmigrants per simulation, marked by the diamonds. The right y axis indicates the number of big outmigrants (length > 5 cm), represented by the square markers. Regression lines are shown with the regression equation and R^2 value. The multiple points for each x value occur because each limiting factors scenario is executed for eight parameter combinations that represent the model's parameter uncertainty.

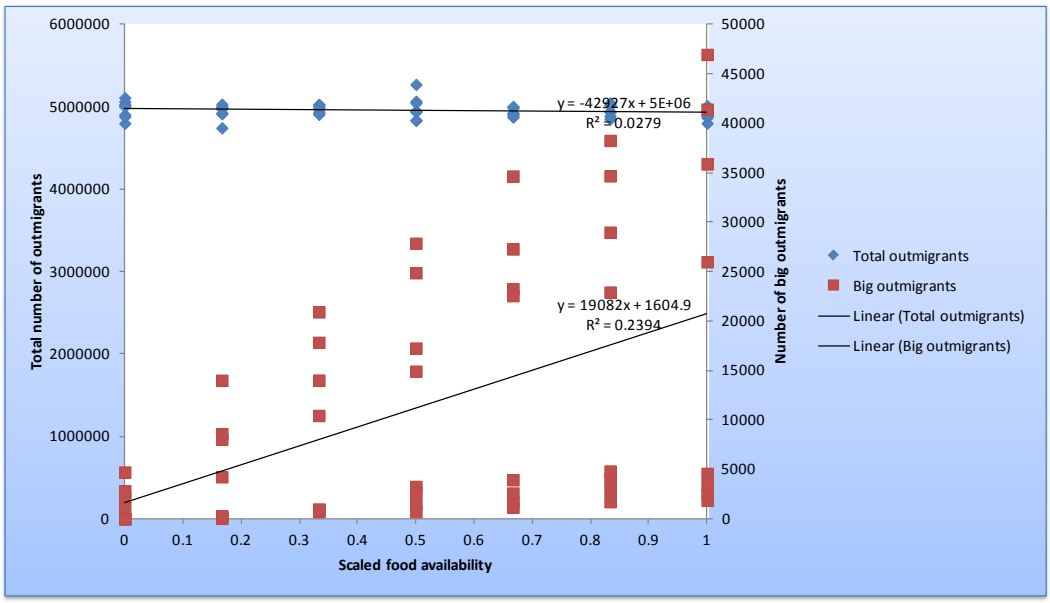


Figure 32. LFT results for food availability. Format is the same as in Figure 31.

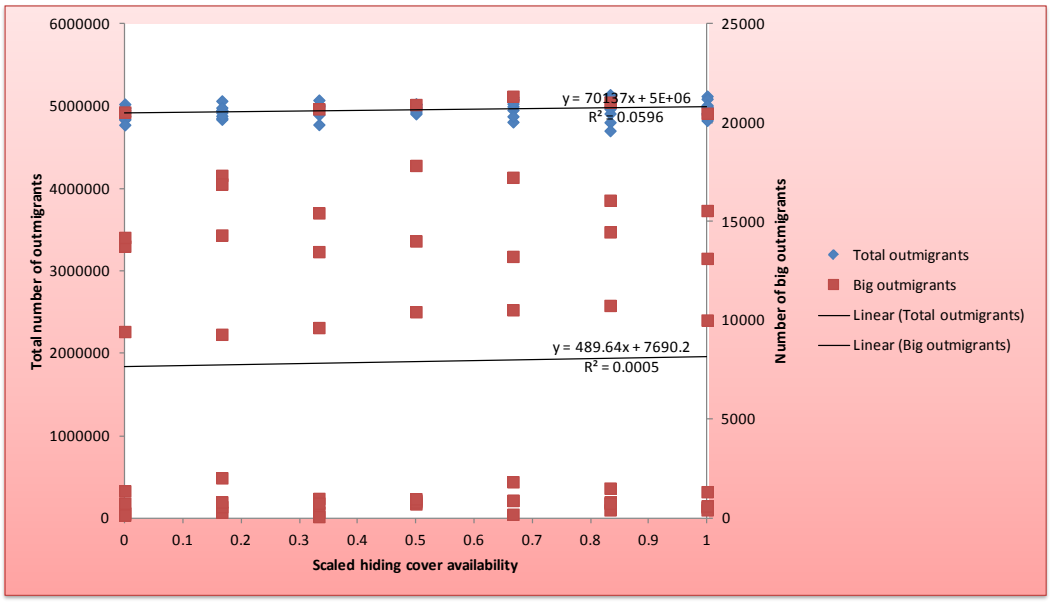


Figure 33. LFT results for hiding cover availability. Format is the same as in Figure 31.

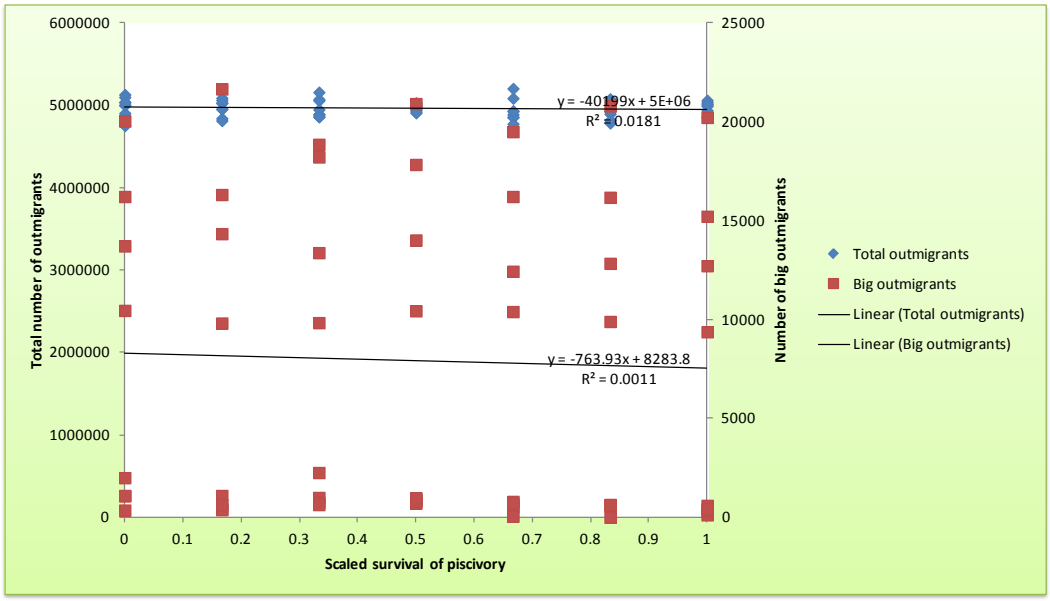


Figure 34. LFT results for piscivory risk. Format is the same as in Figure 31. The x axis represents *survival* of piscivory, with higher values representing lower risk.

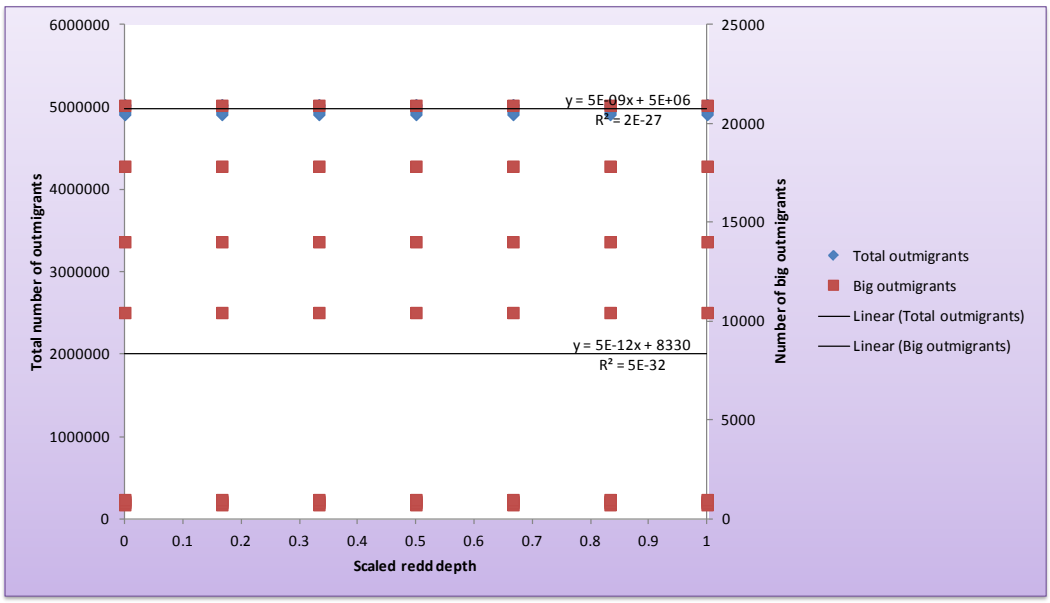


Figure 35. LFT results for redd scour. Format is the same as in Figure 31. The x axis represents redd depth; higher values represent lower risk of redd scour.

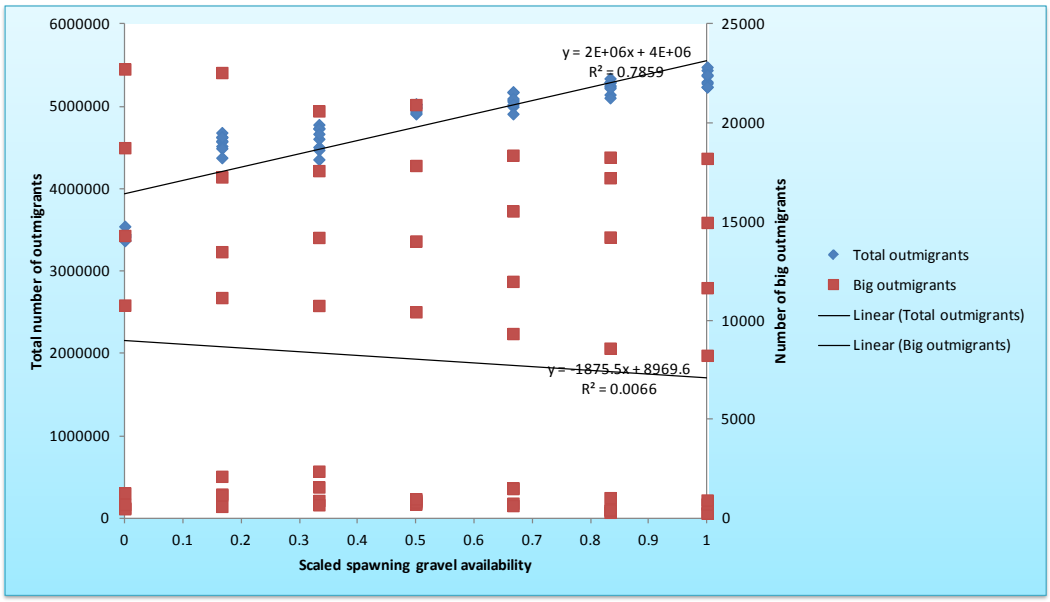


Figure 36. LFT results for spawning gravel availability. Format is the same as in Figure 31.

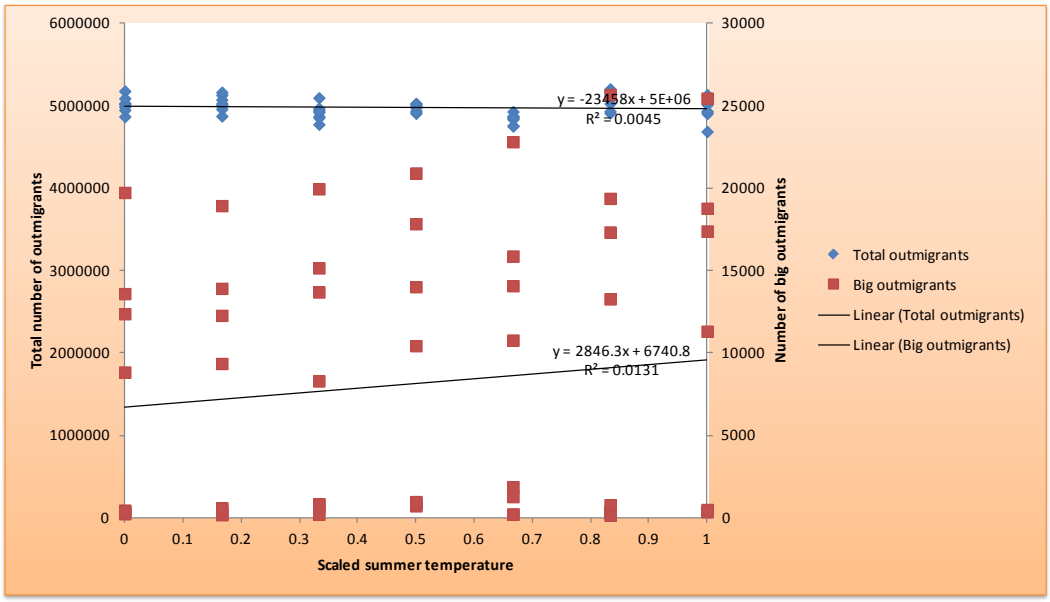


Figure 37. LFT results for summer temperature. Format is the same as in Figure 31.

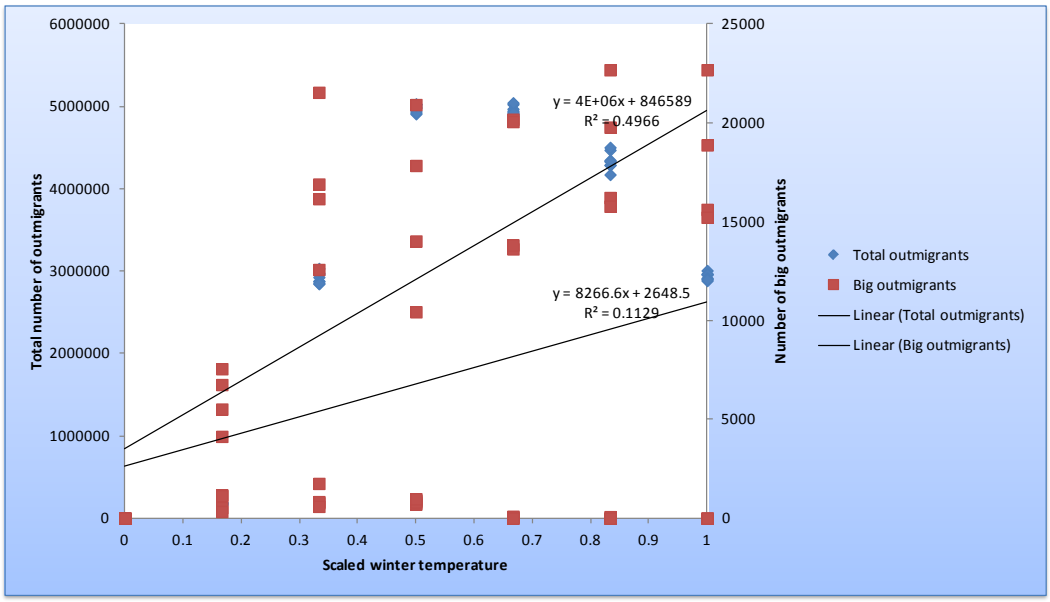


Figure 38. LFT results for winter temperature. Format is the same as in Figure 31.

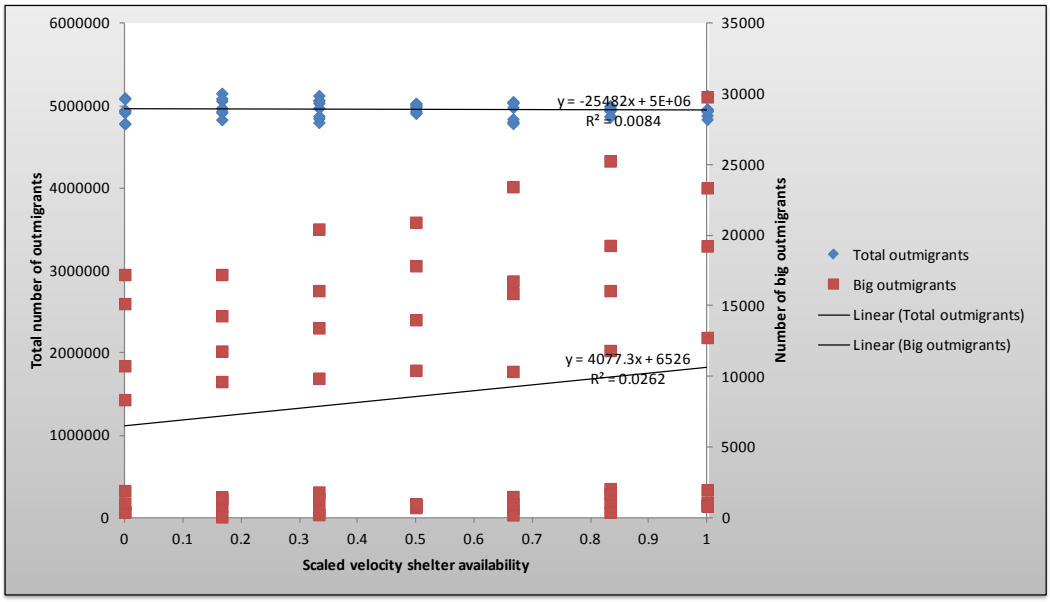


Figure 39. LFT results for velocity shelter availability. Format is the same as in Figure 31.

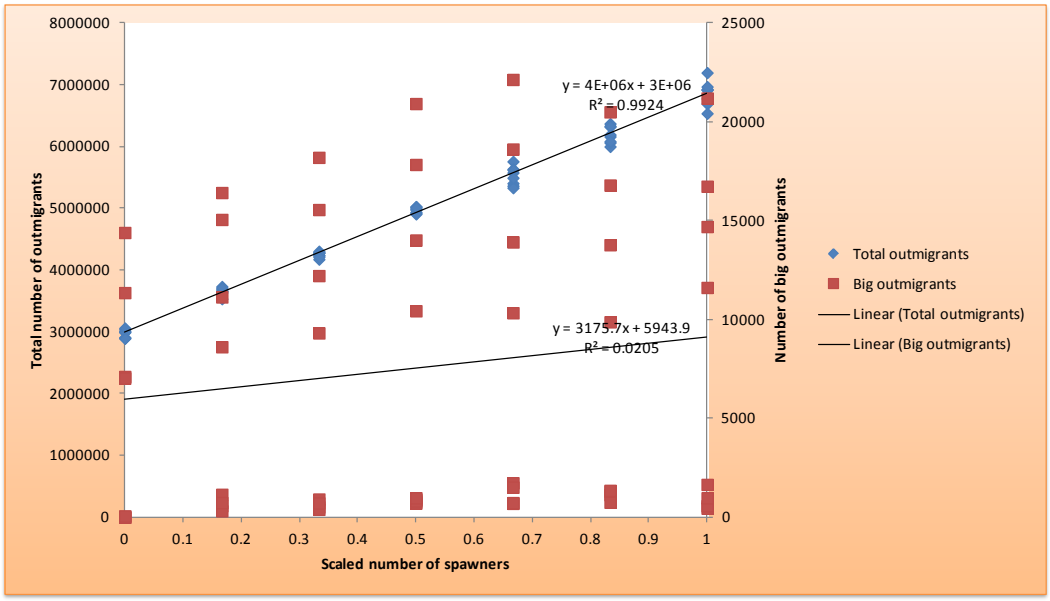


Figure 40. LFT results for number of spawners. Format is the same as in Figure 31.

6.4 Conclusions concerning limiting factors

The LFT and limiting factors analysis of the Clear Creek application were successful in several ways. The analysis indicated several factors that seem to be especially important for production of juvenile salmon. Spawner abundance and gravel availability had strong effects on the total number of outmigrants, while the number of larger outmigrants was most strongly affected by food availability, with velocity shelter availability also important.

The analysis was also successful in that it identified a factor with unexpected and complex effects. Winter temperature was predicted to have very strong effects, but these effects resulted largely from the prediction by *inSALMO* that substantially lower winter temperatures would produce very low egg survival. Small changes in winter temperature were predicted to have little effect. While low-temperature egg survival simulations are uncertain, this result indicates that any management action that might strongly reduce winter temperatures should receive stronger scrutiny than it might normally get.

Another success of the analysis was in identifying factors that seem relatively unimportant. These factors include some most often give highest attention: flow and summer temperature. Piscivory risk had surprisingly low effect (for reasons discussed in Section 4.4.2). Redd scour had no effect, but very likely would have in years with higher winter flows.

This example application illustrates the importance of examining the LFT's graphical results in addition to the summary statistics. Several factors had nonlinear responses not well represented in the statistics but relatively clear from the graphics. Higher numbers of scenarios per experiment help clarify these more complex responses.

7 Development of *inSALMO* Input from Instream Flow Study Data

7.1 Background and objectives

This report section describes methods for developing input for *inSALMO* reaches from data available from USFWS instream flow studies. The work was conducted under project subtask 4.3, which supported integration of *inSALMO* with other agency studies.

The USFWS Sacramento Fish and Wildlife Office has conducted detailed habitat and hydraulic modeling studies at a number of sites throughout Clear Creek as part of their instream flow studies (e.g., USFWS 2011). While these studies did not collect exactly the data required by *inSALMO*, they did assemble the same general kinds of information: hydraulic simulation of a specific representative reach and spatial observations of key habitat variables. The studies measured streambed and bank topography and observed substrate and habitat types at each point in a dense mesh throughout the reach. The River2D hydraulic model was then parameterized for the site and used to simulate depth and velocity at each point in the mesh. Key differences between the USFWS instream flow study products and *inSALMO* input are:

- The USFWS hydraulic modeling was generally conducted at a finer spatial resolution (smaller cell size) than is useful for *inSALMO*.
- The USFWS habitat observations included substrate type and cover availability at points, not the cell habitat variables used by *inSALMO*.
- The USFWS data did not directly evaluate availability of velocity shelters for drift-feeding fish, a key cell variable in *inSALMO*.

The objective of the work described here was to develop methods for interpreting and transforming the USFWS instream flow study products into *inSALMO* input. The methods were developed by creating input for inSALMO for one of the USFWS sites, the Upper Isolation site in the same lower alluvial segment of Clear Creek as the two reaches developed specifically for *inSALMO* (Section 4).

These methods were intended to be general and reproducible so that they can be applied to additional instream flow study sites efficiently. However, we anticipate that the approach explored here will evolve and improve with further application. Using these methods requires some expertise with GIS and the River2D hydraulic model, as well as familiarity with *inSALMO*'s input.

7.2 Data interpretation methods

This section simply lists the steps taken to develop the inputs needed by *inSALMO* to represent a reach. (These inputs are described in full in Section 3 of Appendix A; input file formats are described in Section 5 of Appendix B.) The overall process includes extracting information needed to delineate model cells from River2D, creating the cells, and then calculating hydraulic and habitat variables for each cell.

Much of this process is conducted in GIS. We used the ArcMap software and a separate proprietary add-on to ARC called ET GeoWizards (www.ian-ko.com); however, the same analyses undoubtedly could be conducted using other GIS software. In particular, the full ArcGIS package probably contains the same kinds of tools that ET GeoWizards provides.

7.2.1 Extraction of information from River2D

The first phase is extracting the information needed to create *inSALMO* cells from the River2D hydraulic model output files. The River2D results used are output files for at least two flows: a base flow and a high flow. The *inSALMO* cells are laid out to represent habitat patches that are relatively homogeneous at typical base flows, but must also represent areas that are

submerged only at high flows. Therefore, these steps need to be conducted at least twice, once for a typical base flow and once for a high flow (see Section 7.2.2).

1. The flow-dependent variables required from River2D for *inSALMO* are: depth, velocity magnitude, x discharge intensity, and y discharge intensity. These are obtained from the .cdg file for each flow. The constant variables, obtained from the .chi file, are cover code and substrate code.
2. Open the .cdg file for the desired flow in River2D. Under the Display tab, select "contour/color". In the contour/color window, use the drop-down "Color/Contour Parameter" menu to select the parameter of choice (depth, velocity, x discharge, etc.). This selection determines which values will be extracted.
3. To extract substrate and cover, the River2D .chi files need to be loaded. Under the options tab, select "habitat options" and then "discrete channel index interpolation". Then under the Habitat tab select "Load Channel Index File" and load the appropriate .chi file. Once a .chi file is loaded, the Channel Index selection in the Color/Contour window will display the cover or substrate (whichever .chi file was most recently loaded) values and these can be exported to a comma-separated values (csv) file (see next step).
4. In the main River2D window, scroll over the river reach and observe the values in the lower left corner. These values should be x, y, and the parameter selected in the previous step. Under the Display tab, select "Dump nodal csv file" or "Dump grid csv file"; this choice depends on whether values at each mesh node or at regular grid points are desired. We use mesh nodes in the following methods, but the grid alternative deserves exploration. Save the csv file.
5. The csv files do not have headings, and adding column headers will make data easier to track when importing it to ArcMap. The columns are: (1) node ID, (2) easting (x), (3) northing (y), and (4) output value.
6. The River2D results are in a local coordinate system with units of meters. *inSALMO* does not require coordinates to be in any particular projection or system, but converting the River2D coordinates to a standard projection facilitates analyses such as overlaying the study site with aerial photography. We converted the USFWS River2D coordinates to California State Plane Zone 1 by multiplying the northing (y in River2D) and easting (x in River2D) by 39.37/12 then adding 2062000 to the northings and 6436000 to the eastings.

7.2.2 Delineation of cells

In this phase, the USFWS data are imported to GIS and used to support the delineation of cells for *inSALMO*. The objective is to create a mesh of cells that includes all habitat that is available to fish, ideally even at flood flows; captures important changes in habitat over space while each cell is relatively homogeneous; and includes no more cells than necessary.

The approach is to view the habitat data to see how it varies over space, and then manually create the cells as a polygon mesh. The cell mesh can be created from a triangulated irregular network (TIN) representing depth, but velocity is also an important variable used to delineate cells. Cells can have any number of sides.

The following steps are all conducted in GIS except as noted. The steps need to be conducted at least twice. Cells should be delineated to best represent habitat under the most common or important flows, which are typically "base" flows such as minimum reservoir releases during seasons most important for the fish being modeled. However, cells must also be delineated for areas submerged only at high and flood flows. Therefore, these steps must be followed first using River2D results for a typical or especially important base flow, to delineate the cells submerged at such flows. The second iteration uses a flow high enough to submerge the entire River2D mesh and delineates cells in the higher-elevation areas that not covered at

the first flow. Even more flows may be necessary if the River2D mesh includes much of the floodplain.

1. Import the previously extracted River2D data to GIS. In ArcMap, open the file tab and select "Add Data>Add XY Data". Browse to the csv file being imported and select it. In the X and Y drop-downs, select the appropriate columns of the csv file. The Z drop-down can be left at "none". Once the table is imported, export it to a shapefile.
2. Open the attribute table of the imported data and take note of the attributes. To make the analysis easier, add an ID field and use the field calculator function to make a unique string of integers to represent the data (e.g. the FID+1).
3. Displaying velocities as arrows scaled in size by magnitude and pointing in the flow direction assists in habitat cell delineation. Use the x and y discharge intensity (Qx and Qy), depth (D), and velocity magnitude (V) output files from River2D to produce a file that contains the direction and magnitude of velocity vectors that is then imported and displayed in ArcMap:
    a. In Excel, create a worksheet that contains columns with Qx, Qy, D, and V from the River2D results.
    b. Calculate the direction of the vector. Divide Qx and Qy by the absolute value of D to calculate Vx and Vy. Take the arctangent of Vx and Vy (using the Excel function "=ATAN2(Vy,Vx)") to get the angle of V in radians. Calculate the angle in degrees by using the Excel function DEGREES.
    c. Create an ArcMap input file with the node coordinates, V, and the angle in degrees.
    d. Import these velocity data to ArcMap, export them as a shapefile, and change the display in the "advanced symbology" dialog to rotate the arrow symbol for each node by the velocity angle and scale it by velocity magnitude.
4. Display the depth nodes as a surface (TIN) with elevations discretized appropriately. From this depth TIN, create a TIN domain polygon shapefile.
5. Create the *inSALMO* cells by splitting up the TIN domain polygon. This step should be conducted by someone familiar with how habitat affects fish in *inSALMO*, thinking especially about newly emerged juvenile salmon. Considering spatial variation in velocity as well as depth, use the "split" tool in ArcGIS to add the cells. (If available, high-resolution aerial photography overlaying the TIN can also help determine cell boundaries.)

7.2.3 Calculation of cell habitat variables

In this phase, values are assigned to each *inSALMO* cell for the habitat variables that *inSALMO* treats as constant: the fraction of the cell providing velocity shelter (*cellFracShelter*), a characteristic distance to hiding cover (*cellDistToHide*), and the fraction providing spawning gravel (*cellFracSpawn*). These values are derived from the River2D output, but the general problems are that (a) these variables are not directly represented in the USFWS data, and (b) there are typically many River2D nodes, each with its own habitat variable values, within each of the *inSALMO* cells. (It is also possible that some cells may contain only one or no River2D nodes; we do not consider this possibility here because the USFWS meshes used at Clear Creek were very dense.)

The following steps are used.

1. Create Thiessen polygons around each of the River2D nodes. The polygons define the space that each node represents. Thiessen polygons can be created using ArcToolbox, under "Analysis Tools/Overlay". ET GeoWizards has a Thiessen building tool under the "Surface" drop-down. If there is an option to maintain the attributes of the point input, use it. Once the Thiessen polygons are developed, check to make sure

their attributes include the ID attribute that was added to the point shapefile. If not, determine how to correlate the point shapefile ID to the Thiessen polygon ID.

2. Calculate *cellFracSpawn* as the fraction of each cell that is within a Thiessen polygon for a node that has a USFWS substrate code corresponding to gravel suitable for salmon spawning. We used substrate codes 1.2 and 1.3, which represent medium to large gravel with diameter 1-3 inches (Table 1 of USFWS 2011). This calculation can be conducted using these methods:
    a. Convert the habitat cells into polylines and split the Thiessen polygons by the habitat cell lines.
    b. Calculate the areas for the split Thiessen polygons
    c. Use "Select by Attribute" to identify the Thiessen polygons that have the suitable substrate codes, and export them.
    d. Use the "Spatial Join" tool in ArcToolBox (under "Analysis/Overlay") to associate the Thiessen polygons with suitable substrate codes with their habitat cells.
    e. Export the Attribute Table and bring it into Excel. Create a Excel pivot table to determine the area of spawning substrate for each habitat cell. Those areas can then be divided by total cell areas to obtain *cellFracSpawn*.
3. Calculate *cellFracShelter* for each cell. This step is the most complex and indirect because velocity shelter is least directly represented in the USFWS data. Velocity shelter can be provided by large substrate (e.g., downstream of boulders) and by steep velocity gradients in the horizontal dimensions. The method we used is based on the assumption that the dense USFWS mesh of River2D nodes captures such gradients adequately to represent velocity shelter. Hence, the approach quantifies the fraction of a cell that is close enough to a node with substantially lower velocity that the gradient could be used as velocity shelter by a juvenile salmon. Neighboring area is considered velocity shelter if its velocity is less than 40% of a cell's. The approach also considers that velocity shelter is meaningless when velocities are low even without shelter. We used 0.15 m/s as the threshold velocity below which shelter is meaningless; this is approximately the velocity at which the metabolic cost of swimming becomes significant for juvenile salmon in *inSALMO*.
    a. Create a shapefile containing the velocity nodes from River2D, with an attribute for which *inSALMO* cell they are in. Start with the velocity point shapefile and create a Thiessen polygon shapefile from it: create a unique ID code for each node, and then build Thiessen polygons for the nodes. Calculate the area of each such polygon and add it to the shapefile's attribute table. Select all Thiessen polygons that fall within or intersect *inSALMO* cells, and export them as a new shapefile.
    b. In this new shapefile, use the Spatial Join tool to add an attribute that identifies the *inSALMO* cell associated with each point (using the "one to many" and "contains" options in the spatial join tool). Points that are on a cell boundary will have multiple cell ID codes for this attribute. The resulting shapefile is referred to as SF1. Export the attribute table to Excel.
    c. Select by location the Thiessen polygons for velocity nodes that are within or intersect each *inSALMO* cell. Add an attribute field and calculate the area of these polygons, and export as a new shapefile.
    d. Select by attribute the Thiessen polygons that have velocity values greater than 0.15 m/s and export them as a new shapefile. Use the Spatial Join tool to identify all the polygons adjacent to these selected Thiessen polygons (in the Spatial Join tool, "Target" the polygon shapefile of values greater than 0.15 m/s and "Join" to the polygon shapefile of all Thiessen polygons created in step c, above). Select the "one to many" and "intersect" options. The resulting shapefile is referred to as SF2. Inspect the attribute table and identify which fields (there will be duplicates of velocity, area, and ID) are as-

sociated with which file (adjacent polygons and polygons with values greater than 0.15 m/s). Export the attribute table to Excel.

e. Put the attribute tables for SF1 and SF2 in one Excel workbook. Create a new worksheet with the following columns.
    i. Thiessen polygon ID (Threshold IDs): IDs of all the polygons with velocities greater than 0.15 m/s, pasted from the attribute table of SF2.
    ii. Habitat cell containing node of Thiessen polygons (the cell that contains the node of the Thiessen polygon in col. 1). The value is determined using a lookup table (function VLOOKUP), and uses SF1 to determine what cell the nodes in col. 1 are in.
    iii. Velocity value of the polygon in col. 1. This can be pasted from the attribute table of SF1.
    iv. Neighbor IDs: the ID codes for all polygons adjacent to those in col. 1. These can be pasted from the attribute table for SF2.
    v. Cell containing neighbor polygon: the ID of the cell containing the polygon in col. 4. The value can be determined with the lookup function used in col. 2.
    vi. Neighbor Thiessen polygon velocity: the velocity values of all polygons adjacent to those in col. 1. This can be pasted from the SF2 attribute table.
    vii. Neighbor Thiessen polygon area: the area of all polygons adjacent to those in col. 1. This can be pasted from the SF2 attribute table.
    viii. 40% of polygon velocity: this is the value of col. 3 multiplied by 0.4.
    ix. Is the neighbor polygon velocity shelter? This is a true-false value that is true if col. 6 is less than col. 8. However, this column must also exclude as velocity shelter polygons in cells that are dry at the flow being analyzed. One way to do so is to create a separate list of cell IDs for cells that are not dry and check col. 5 against it.
    x. Neighbor polygon shelter area: if col. 9 is true then this is equal to col. 7; otherwise it is zero.
    xi. Unique ID for polygons providing shelter: if col. 9 is true, then this is col. 4 with the letter "S" concatenated to it.
    xii. Habitat cell associated with shelter: if col. 9 is true, then this is col. 5.
    xiii. Combined cell and neighbor ID: if col. 9 is true, this is the concatenation of the ID codes in cols. 11 and 12.
    xiv. The number of times the combination of cell and polygon occur: if col. 9 is true, then this is the number of times the value in col. 13 of the current row occurs throughout col. 13. The Excel function "countif" can be used.
    xv. Shelter area divided by the number of occurrences: If col. 9 is true, then this is col. 10 divided by col. 14. (If col. 9 is false, its value can be set to the Excel function "=NA()".)

f. Now an Excel pivot table can identify all the *inSALMO* cells that are associated with shelter (col. 12) and their shelter area (sum of col. 15). These values are divided by cell area to get *cellFracShelter*.

4. Calculate *cellDistToHide* for each cell as the average distance to any kind of hiding cover, over 100 random points within the cell. Hiding cover was defined as a River2D node with a USFWS cover code representing hiding cover; we included all codes of 3.7 or higher, representing most kinds of cover noted by the USFWS (Table 2 of USFWS 2011). The following methods can be used:
    a. Create 100 random points within each polygon, using a tool such as ET GeoWizards' "random point generator within polygon". The resulting random

point shapefile should contain an attribute for the ID of the polygon that each random point is within.

b. Calculate the distance from each random point to the nearest River2D node with hiding cover. We generated a separate point layer for such nodes, then used a point distance tool of ET GeoWizards.
c. Export the random points, with their distance to nearest hiding cover attribute. In Excel, use a pivot table to determine the average distance to cover for the 100 random points in each *inSALMO* cell. These averages are used as *cellDistToHide*.

7.2.4 Calculation of cell hydraulic variables

inSALMO requires a lookup table containing the average depth and velocity magnitude of each cell, at many flows spanning the range of flows to be simulated. To build this table for a reach, the River2D hydraulic output (.cdg) file for each flow must be opened and cell-average depth and velocity values determined.

There are several ways that cell-average depth and velocity can be calculated. The simplest is to simply average the depth and velocity values for all River2D nodes within a cell. Alternatively, the cell area represented by each node can be accounted for by calculating averages weighted by the area of the Thiessen polygon around a node that is within the cell. Methods for these approaches are readily adapted from techniques described in Section 7.2.3 for habitat variables, especially *cellFracSpawn*.

7.2.5 Estimation of shear stress and scour parameter values

inSALMO uses several reach-specific parameters to calculate the probability of redd scour on peak flow days. These parameters depend on geomorphic factors such as channel shape and substrate size. Parameter values should be estimated by someone with expertise in river geomorphology, using standard techniques for relating shear stress to variables such as depth, substrate size, and slope.

River2D results for very high flows—much higher than the base flows that River2D is typically calibrated to for instream flow studies—can be used to estimate shear stress parameters if the hydraulic model calibration works reasonably well at the high flows. If River2D does not work well at potential scour flows, then other information such as one-dimensional flood model results or field observations of water surface elevation must be used. The use of River2D results and an alternative are illustrated in Section 4.2.4.

For the Upper Isolation site we simply used the shear stress parameters developed for site 3A, which is relatively similar geomorphically.

7.3 Example: Upper Isolation Site

The methods described above were developed by applying them to data provided by Mark Gard (USFWS) for the Upper Isolation instream flow study site, which is in Clear Creek's lower alluvial segment and upstream of sites 3A and 3C.

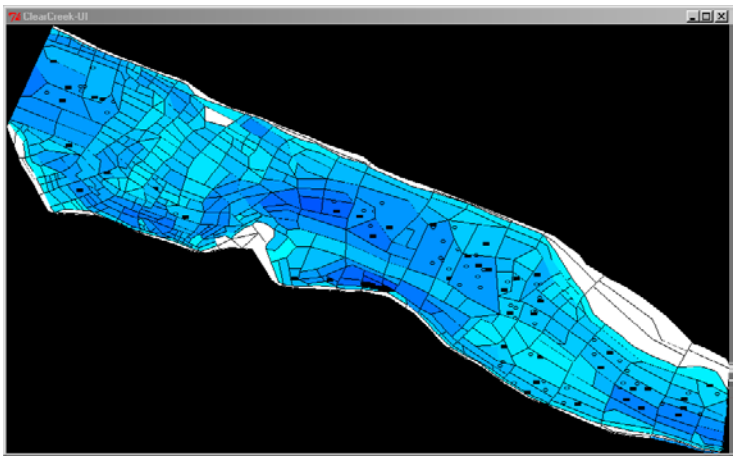


Figure 41. The Upper Isolation site in *inSALMO*. Cells are shaded by depth, with darker cells deeper. Black rectangles are adults, open ovals are redds.

The resulting Upper Isolation reach in *inSALMO* has 480 cells of relatively complex habitat (Figure 41). Before spawning, adults tend to occupy a deep cell near the middle of the reach on the south side. Redds tend to be distributed longitudinally throughout the reach except for a region just downstream (right, in Figure 41) of its upper end. Compared to the other two sites, Upper Isolation has more cells with high availability of velocity shelter; higher distances to hiding cover; and considerably higher availability of spawning gravel (Figure 42).

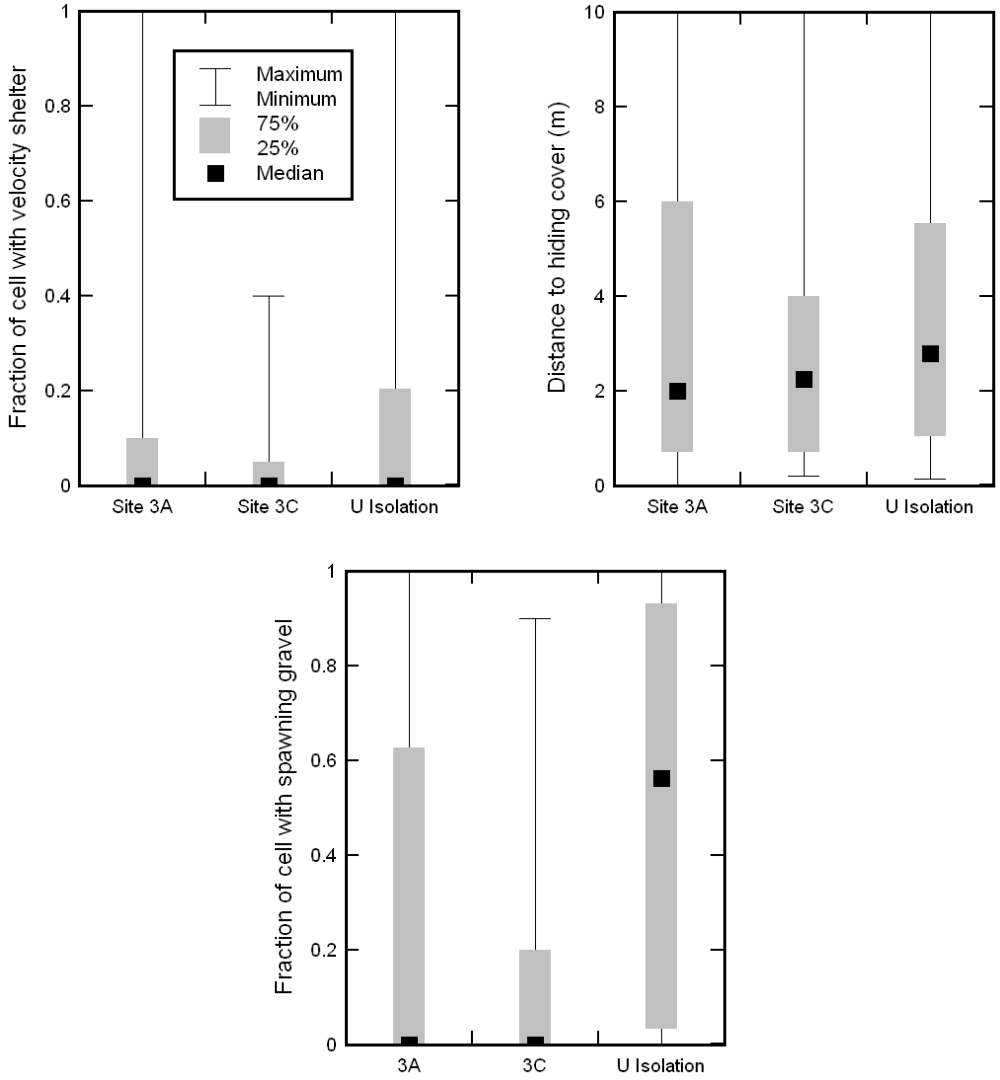


Figure 42. Habitat variable distributions at the Upper Isolation site compared to sites 3A and 3C. The plots show the distributions among cells in velocity shelter availability, distance to hiding cover, and spawning gravel availability. Values are not weighted by cell area, and all cells (including those dry at normal flows) are included.

Preliminary model results indicate that the Upper Isolation reach provides relatively good habitat for spawning and rearing of Chinook juveniles. Simulations for water years 2005-9 indicated that Upper Isolation, despite being only about 40% of the length of sites 3A and 3C, and so being initialized with only 40% of the spawners, produced almost 60% of the large juveniles as site 3A (Figure 43).

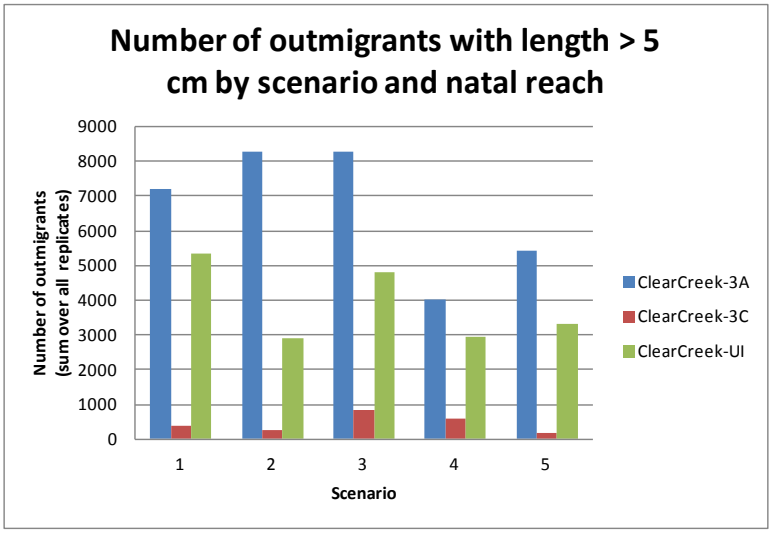


Figure 43. Simulated number of large (length > 5 cm) outmigrants for Upper Isolation site ("Clear Creek-UI") in comparison to sites 3A and 3C. Scenarios 1-5 refer to water years 2005-2009. On average over all these years, Upper Isolation produced 58% as many large outmigrants as 3A and almost 9 times as many as site 3C.

8 Model Research and Development Needs

8.1 Objectives and methods

Project Task 6 is to identify priorities for future research and development (R&D) to improve *inSALMO* and its application to management of Central Valley Chinook salmon populations. This section serves as the R&D needs document product of Task 6.

We identified R&D needs in the following three groups.

Improvements to the Clear Creek application. These are actions that could be undertaken with relatively little additional effort and existing information to further test or improve the model and its pilot application to Clear Creek. These were identified primarily by considering two questions:

- What information and data do we understand to be available, and
- How could we use that information to improve the model application, and how important would those improvements be?

Short-term, general improvements to *inSALMO*. This project identified several relatively simple improvements that could make *inSALMO* more useful in general, but were beyond the project's current scope.

Longer-term field- and model-based research. These studies would test and improve model assumptions that proved particularly important in this project. Long-term R&D needs were identified primarily by considering the important uncertainties remaining in the model and the process of making management decision supported by *inSALMO*. These needs were based on questions such as:

- What parameters and processes are indicated by the sensitivity analysis and limiting factors analysis to be particularly important, and also relatively uncertain?
- Are there promising ways to reduce these uncertainties?
- What important river management considerations are not yet represented in *inSALMO*?

The following three subsections discuss these three kinds of future R&D.

One research topic not included here is the relative importance of small vs. large outmigrants. At Clear Creek, the vast majority of juvenile Chinook move downstream as very small fry immediately after emerging from redds, while a small minority stay and grow before outmigration. The model indicates that management to increase the number of total outmigrants can conflict with management to increase the number that stay and grow. A clearer understanding of which size category is most important to adult abundance could strongly influence how *inSALMO* results are interpreted and used to support decisions. However, this question is also of broad management importance, and modeling does not seem like the best approach for answering it. Instead, empirical studies such as that of Miller et al. (2010) and field studies that track migrants downstream appear most promising for providing conclusive evidence. (A related question is addressed in Section 8.4.5.)

8.2 Improvements to the Clear Creek application

8.2.1 Model additional reaches

The initial application of *inSALMO* 1.0 reported here used two reaches to represent one part of Clear Creek. Modeling these two reaches produced important conclusions, including that

the two reaches produced quite different results (Section 5.1). Another issue raised by the initial application is that our ability to calibrate and test the model against rotary screw trap data (Section 4.5) is limited because our two reaches may not adequately represent the stream reach over which the fish sampled by the trap are produced. One key model result, the number of relatively large outmigrants, was found to vary with how many reaches were simulated (Section 4.4.2).

As part of subtask 4.3, we developed a process for developing model reach input from USFWS instream flow modeling studies. This process requires substantial effort but allows us to produce additional model reaches without extensive field work. The USFWS studies are available for approximately 10 sites in the lower alluvial segment of Clear Creek, plus sites in upper Clear Creek.

Adding more reaches to the Clear Creek model could have several benefits. More reaches on the lower alluvial segment would improve the degree to which the *inSALMO* application represents this section, making the model more supportable for management decision support. These new reaches would also allow us to evaluate how sensitive model results are to the variety as well as number of reaches included. Adding sites on Clear Creek above the lower alluvial segment would improve how the model represents the creek for the species and races (spring Chinook; steelhead) that tend to spawn farther upstream.

8.2.2 Model additional Chinook races

This project simulated fall-run Chinook salmon. A late-fall race also spawns in the lower alluvial segment of Clear Creek that contains our two study reaches, but is much lower in abundance than fall-run. Spring-run Chinook also occur in Clear Creek but tend to hold and spawn well upstream of the sites currently modeled.

Adding Chinook salmon races other than fall run to the Clear Creek application should be relatively straightforward, at least for races that use the same habitat reaches currently in the application. The software changes are trivial, so the major effort would be in assembling the input—especially, how many spawners arrive when and where—for the new races. We have not investigated the availability of such information. The current version of *inSALMO* was designed to be capable of simulating races that hold for extended periods before spawning, though this capability has not been tested.

The primary advantage of adding the other races that use Clear Creek would be in allowing use of the model to identify any conflicts or tradeoffs among the races: are there management actions (changes in flow, etc.) that might benefit one race but harm another? Another potentially important advantage is that there could be interactions among the races, especially competition for habitat. In that case, including all the races could improve the model by allowing it to represent those interactions.

8.2.3 Test and evaluate juvenile habitat selection

One of the key results of the test application to Clear Creek is that production of larger juveniles depends on availability of good foraging habitat for newly emerged juveniles (Section 5.1.3). The validity of this result could be tested in part by evaluating how well *inSALMO* predicts habitat use by such juveniles. Our understanding is that some habitat use data are available but we have not had the opportunity to review it for usefulness in testing the model. Collecting additional data specifically for this purpose should be relatively straightforward and inexpensive.

8.2.4 Test and calibrate redd scour predictions

The redd scour model and the parameter values developed for Clear Creek (Section 4.2.4) predict that scour should start to occur (> 1% of gravel disturbed to a depth > 20 cm) at flows of 100 m^3/s (3500 cfs) at 3C and at 65 m^3/s (2300 cfs) at 3C. These values are in the range

reported verbally by USFWS staff for the onset of scour at Clear Creek. We have not reviewed available data to determine the extent to which the scour submodel could be tested quantitatively. The years we have used most in model runs (2007-9) produced no simulated scour events.

8.3 Short-term general improvements to *inSALMO*

8.3.1 Represent spawner selection among reaches

Version 1.0 does not let spawners move among reaches when selecting spawning habitat; they are forced to spawn in the reach where they were put by the model. This design was chosen to facilitate controlled experiments: the number of spawners per reach—an important factor determining model results—does not vary unless the user makes it vary. The design also lets the user force the number of spawners in each reach to match observed values.

A disadvantage of this design is that it does not let *inSALMO* predict how habitat differences among reaches (e.g., addition of spawning gravel to one reach) affect the number of adults that spawn in each reach. If spawners could select where to spawn from among several reaches, the model would represent such effects.

The capability to let spawners select among reaches probably can be added with a simple switch that users control for each model run. However, before such a switch is added to the standard version of the model its effects need to be tested and analyzed to determine its usefulness. Guidance on when to use each alternative would need to be developed.

8.3.2 Model effects of spawning gravel quality

Version 1.0 of *inSALMO* represents the effects of spawning gravel quantity but not quality. Gravel quality is clearly a management concern; sedimentation and infrequent flushing flows can degrade quality and potentially affect such processes as spawning site selection and redd survival. Gravel quality can have a variety of effects that can be complex and challenging to model (Kondolf 2000). These complex effects have not yet been included in *inSALMO* because it and predecessor models were not previously focused as strongly on spawning.

The addition of spawning gravel quality effects to *inSALMO* is unlikely to be trivial. Key questions include:

- Which of the various effects of gravel quality are important enough to include?
- What measures of gravel quality are useful: observable in the field or even predictable with models, and clearly related to processes such as redd site selection and egg survival?
- What are useful submodels for how the gravel quality measures affect salmon?

8.3.3 Model differences between wild and hatchery spawners

Fisheries agency participants in this project expressed concern about potential differences between hatchery-reared vs. wild spawners, and how these differences might affect juvenile salmon production. Hatchery-produced spawners might differ in, for example, fecundity, energy reserves, or some measure of fitness in their eggs and fry. Adding such differences to the model could be technically easy but (as with spawning gravel quality) should be preceded by careful consideration of what differences are well-documented and clearly meaningful to the model's objectives. We have not investigated this question.

8.4 Longer-term field- and model-based research

8.4.1 Improve representation of fry habitat selection and outmigration

The pilot application of *inSALMO* makes it clear that the fate of fry immediately after emergence from their redds is very important to model results. The model and screw trap data

agree in showing that the vast majority of newly emerged salmon migrate downstream very soon. However, it is not clear that the mechanisms causing this rapid outmigration in the model—lack of suitable rearing habitat near the redds—are the only real cause. Other possibilities are that newly emerged fry simply do not have the swimming ability to reach habitat with suitably low velocities, and that immediate outmigration is an evolved adaptive behavior.

Carefully controlled field studies could help determine which mechanisms are important in causing new fry to migrate downstream rapidly vs. staying in their natal reach. Potential field techniques include comparing the timing of outmigration to that in streams with more low-velocity habitat. Direct observation of newly emergent fry may provide useful evidence. Studies to clarify the relative fitness of small vs. large outmigrants (discussed in Section 8.1) would also be relevant to the hypothesis that early outmigration is an evolved adaptive behavior.

8.4.2 Test outmigration decision submodel

Our submodel for how juvenile salmon decide when to migrate downstream (Section 4.2.5 of Appendix A) is simple yet produces a variety of behaviors that depend on fish size, growth, and survival probability. The submodel, though, is theoretical and has not been tested extensively against empirical observations. Testing the submodel (in comparison to alternative approaches) would improve confidence in the ability of *inSALMO* to predict how outmigration varies with habitat conditions.

The outmigration submodel would be very difficult to test directly by controlling habitat variables and observing behavior at the individual level. However, this kind of behavior theory in individual-based models can be tested via the "pattern-oriented modeling" strategy (Railsback 2001; Grimm et al. 2005). This strategy contrasts alternative theories for a key behavior by how well each causes an individual-based model to reproduce a variety of empirically observed patterns. This pattern-oriented theory development strategy has been used to confirm other behavior submodels used in *inSALMO* and related *inSTREAM* models (Railsback and Harvey 2002; Railsback et al. 2005).

The outmigration submodel might best be tested by how well it reproduces patterns in outmigration observed across a variety of sites, because the basic pattern observed at Clear Creek (early outmigration by a high percentage of juveniles) is consistent over time and between the two reaches. However, the different percentages of juveniles remaining and growing at sites 3A and 3C could be a pattern against which alternative outmigration theories could be tested. Our submodel could be tested against both simpler and more elaborate theories (e.g., that of Satterthwaite et al. 2009).

8.4.3 Model steelhead

Steelhead are an important component of the anadromous salmonid communities and management priorities in many California streams. Adapting *inSALMO* to other salmon species and races is not expected to be difficult (Section 8.2.2), and steelhead could just as easily be added *if* the diversity of steelhead life histories was ignored. However, conserving this diversity is often a key management concern. It appears that two alternative approaches could be taken to modeling steelhead.

The simplest approach would be to treat steelhead as we do Chinook, assuming that all adults spawn and that all juveniles eventually migrate downstream. This assumption may accurately represent a large percentage of real steelhead, and such a model may be very useful for many management decisions. It could, for example, represent the effects of flow, temperature, and spawning gravel on steelhead spawning and production of age-0 juveniles. However, this method could not predict how management alternatives affect other more complex steelhead life histories.

This simple alternative is expected to be relatively straightforward to implement. Only minor changes to the software should be needed. Input that would need development includes: model reaches that represent steelhead spawning and rearing areas; the number, size, and timing of adult steelhead arrival; and values for the parameters by which steelhead are believed to differ from Chinook.

The second approach is to include the life history diversity of real steelhead and the decisions they make among life histories. This approach would allow *inSALMO* to predict how river management would affect the number of steelhead that migrate downstream at different ages and (perhaps) the number of adults that remain in fresh water vs. attempting to return to the ocean. Conceptual approaches for modeling these decisions potentially suited for *inSALMO* have been developed (e.g., Grand 1999; Satterthwaite et al. 2009). The pattern-oriented theory development strategy (Section 8.4.2) should be used to compare and validate such approaches. At least in principle, this more complex approach to modeling steelhead would have the advantage of being useful for management of their life history diversity.

The potential benefits of the second alternative are likely to come with important costs. Our experience with putting additional adaptive behaviors in models like *inSALMO* (especially, the addition of diel feeding vs. hiding decisions; Railsback et al. 2005) has been mixed. The model revisions can be complex and difficult to design and implement; the resulting model can succeed in displaying a wide variety of realistic behaviors; but the model's additional complexity can make it more difficult to calibrate and understand. The choice between these two alternatives deserves careful consideration, especially of what management decisions the model is most needed to support.

8.4.4 Evaluate the relation between cover and predation risk

The application of *inSALMO* to Clear Creek indicates that availability of cover can have important benefits for the production of large juveniles (Section 6.4). However, almost all Central Valley salmon streams have populations of piscivorous fish that appear to use the same kinds of cover, possibly to hunt juvenile salmon. Hence, it is not clear whether additional cover is beneficial for salmon, or whether much of it is avoided because of predation risk, or what kinds of cover could be "sink" habitat that attracts salmon but exposes them to high predation risk.

To our knowledge, the relation between cover and piscivory risk has not been studied adequately to make informed decisions about the actual benefits of adding cover. A better understanding of this relation seems attainable using techniques such as tethering prey fish to evaluate relative predation rates in different kinds of habitat and using direct or indirect (e.g., hydroacoustic) observation of predator fish locations. An improved understanding of this relation would have very important benefits to many management decisions, not just those based on *inSALMO*.

8.4.5 Model survival and rearing in the delta

In our model results as well as screw trap data, most juvenile salmon migrated downstream shortly after emergence. Even the fry that grew in their natal stream rarely approached a typical smolt size before migrating out. These observations indicate that the fate of outmigrants between their natal stream and the ocean is very important.

The original vision of *inSALMO* was as a full-life-cycle model (Kimmerer and Jones and Stokes Associates 1999), though implementation of that vision lacked the detail necessary to address important management questions at each life stage (Railsback 2005). Version 1.0 of *inSALMO* focuses in more detail on freshwater life stages through downstream migration, but its results strongly indicate that what happens to juveniles downstream between their natal reaches and the ocean is very important. The importance of understanding how management

of the delta (flows, export pumping, sport fisheries, etc.) affects juvenile salmon is widely accepted.

We are not aware of all the modeling efforts focused on this question, but are aware of one past effort that seems promising for development into an individual-based model with an appropriate level of detail for simulating the effect of delta management on juvenile salmon. This effort was undertaken by Annjanette Dodd under a 2005 CALFED postdoctoral research fellowship. The fellowship project was intended to develop a model of juvenile salmon outmigration through the delta that included fish behavior, but it ended up focused on the critical and surprisingly challenging question of how to use the now-voluminous acoustic tag data to test migration models.

The general approach developed by Dr. Dodd appears promising for development into a model similar in capability and detail to *inSALMO* 1.0 but addressing the delta. This approach consists of:

- Modifying an existing particle tracking model of the delta to represent juvenile salmon with behavior. The key behaviors appear to be selecting swimming speeds and directions, and deciding which channel to choose at junctions. The modified model also simulates the acoustic tag detectors now placed throughout the delta.
- Identifying a range of alternative theories for the fish behaviors, starting with the "null" theory that fish act as passive particles. More elaborate theories can assume, for example, that the preference for downstream migration vs. growth increases as fish approach smolt size.
- Identifying a set of measures for evaluating how well simulated acoustic tag observations match actual tag data, as a way to test how well the model represents real fish movement through the delta.
- Conducting simulation experiments to see how well each alternative theory for fish behavior causes the model to fit the observed acoustic tag data.

If the need for a delta outmigration model remains unfulfilled, this approach could be developed as the next major component of *inSALMO*.

9 References cited

Beacham, T. D., and C. B. Murray. 1990. Temperature, egg size and development of embryos and alevins of five species of Pacific salmon: a comparative analysis. Transactions of the American Fisheries Society 119:927-945.

Burner, C.J. 1951. Characteristics of spawning nests of Columbia River salmon. Fishery Bulletin 61, 52:97-110.

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.

Earley, J. T., D. J. Colby, and M. R. Brown. 2010. Juvenile salmonid monitoring in Clear Creek, California, from October 2008 through September 2009. U.S. Fish and Wildlife Service, Red Bluff Fish and Wildlife Office, Red Bluff, CA.

Graham Matthews & Associates (GMA). 2007. Clear Creek geomorphic monitoring project, WY2006 annual report. Report prepared by GMA for Western Shasta Resource Conservation District, Anderson, CA.

Grand, T. C. 1999. Risk-taking behavior and the timing of life history events: consequences of body size and season. Oikos 85:467-480.

Grimm, V., E. Revilla, U. Berger, F. Jeltsch, W. M. Mooij, S. F. Railsback, H.-H. Thulke, J. Weiner, T. Wiegand, and D. L. DeAngelis. 2005. Pattern-oriented modeling of agent-based complex systems: lessons from ecology. Science 310:987-991.

Harvey, B. C., and S. F. Railsback. 2007. Estimating multi-factor cumulative watershed effects on fish populations with an individual-based model. Fisheries 32:292-298.

Healey, M. C., and W. R. Heard. 1984. Inter- and intra-population variation in the fecundity of chinook salmon (*Onchorynchus tshawytscha*) and its relevance to life history theory. Canadian Journal of Fisheries and Aquatic Sciences 41:476-483.

Kimmerer, W. and Jones & Stokes Associates, Inc. 1999. Conceptual framework for the Sacramento River Chinook salmon individual-based model. Draft report JSA 96-170 prepared by Jones & Stokes Associates for the U.S. Fish and Wildlife Service, Sacramento, CA.

Kondolf, G. M. 2000. Assessing salmonid spawning gravel quality. Transactions of the American Fisheries Society 129:262-281.

Mesa, M. G., and C. D. Magie. 2006. Evaluation of energy expenditure in adult spring Chinook salmon migrating upstream in the Columbia River Basin: an assessment based on sequential proximate analysis. River Research and Applications 22:1085-1095.

Miller, J. A., A. Gray, and J. Merz. 2010. Quantifying the contribution of juvenile migratory phenotypes in a population of Chinook salmon *Oncorhynchus tshawytscha*. Marine Ecology Progress Series 408:227-240.

Myrick, C. A., and J. J. Cech, 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.

Newton, J. M. and M. R. Brown. 2004. Adult Spring Chinook Salmon Monitoring in Clear Creek, California, 1999-2002. USFWS Report. Red Bluff, CA.

Petrusso, P. A., and D. B. Hayes. 2001. Condition of juvenile Chinook salmon in the upper Sacramento River, California. California Fish and Game 87:19-37.

Railsback, S. F. 2001. Getting "results": the pattern-oriented approach to analyzing natural systems with individual-based models. Natural Resource Modeling 14:465-474.

Railsback, S. F., R. H. Lamberson, B. C. Harvey, and W. E. Duffy. 1999. Movement rules for spatially explicit individual-based models of stream fish. Ecological Modelling 123:73-89.

Railsback, S., and B. Harvey. 2001. Individual-based model formulation for cutthroat trout, Little Jones Creek, California. General Technical Report PSW-GTR-182, Pacific Southwest Research Station, Forest Service, U. S. Department of Agriculture, Albany, CA.

Railsback, S. F., and B. C. Harvey. 2002. Analysis of habitat selection rules using an individual-based model. Ecology 83:1817-1830.

Railsback, S. F., B. C. Harvey, R. H. Lamberson, D. E. Lee, N. J. Claasen, and S. Yoshihara. 2002. Population-level analysis and validation of an individual-based cutthroat trout model. Natural Resource Modeling 15:83-110.

Railsback, S. F., H. B. Stauffer, and B. C. Harvey. 2003. What can habitat preference models tell us? Tests using a virtual trout population. Ecological Applications 13:1580-1594.

Railsback, S. F. 2005. Individual-based Chinook salmon modeling for the Sacramento River Basin: information and research needs. Report prepared for Jones and Stokes Assoc. and U.S. Fish and Wildlife Service. Lang, Railsback & Assoc., Arcata CA.

Railsback, S. F., B. C. Harvey, J. W. Hayse, and K. E. LaGory. 2005. Tests of theory for diel variation in salmonid feeding activity and habitat use. Ecology 86:947-959.

Railsback, S. F., B. C. Harvey, S. K. Jackson, and R. H. Lamberson. 2009. inSTREAM: the individual-based stream trout research and environmental assessment model. PSW-GTR-218, USDA Forest Service, Pacific Southwest Research Station, Albany, CA.

Railsback, S. F., J. W. Hayes, and K. E. LaGory. 2006. Simulation analysis of within-day flow fluctuations on trout below Flaming Gorge Dam. Technical Report 1012855, EPRI, Palo Alto, California.

Satterthwaite, W. H., M. P. Beakes, E. M. Collins, D. R. Swank, J. E. Merz, R. G. Titus, S. M. Sogard, and M. Mangel. 2009. Steelhead life history on California's central coast: insights from a state-dependent model. Transactions of the American Fisheries Society 138:532-548.

US Bureau of Reclamation (USBR) 2008. Sensitivity of Future Central Valley Project and State Water Project Operations to Potential Climate Change and Associated Sea Level Rise. Appendix R of: Biological Assessment on the Continued Long-term Operations of the Central Valley Project and the State Water Project, report prepared by the USBR Mid-Pacific Region, Sacramento CA.

US Fish and Wildlife Service (USFWS), Red Bluff Fish and Wildlife Office. 2008. Fall Chinook salmon spawning area mapping for the Clear Creek Restoration Program, 2007. USFWS Report. Red Bluff, CA.

US Fish and Wildlife Service (USFWS), Sacramento Fish and Wildlife Office. 2011. Flow-habitat relationships for fall-run Chinook salmon and Steelhead/Rainbow trout spawning in Clear Creek between Clear Creek Road and the Sacramento River. USFWS Report. Sacramento, CA.

Vigg, S., and C. C. Burley. 1991. Temperature-dependent maximum daily consumption of juvenile salmonids by northern squawfish (*Ptychocheilus oregonensis*) from the Columbia River. Canadian Journal of Fisheries and Aquatic Sciences 48:2491-2498.

Appendix A: Model Description

inSALMO Version 1.0 Model Description

Prepared by:

Steven F. Railsback
Lang, Railsback & Associates
Arcata, CA
www.LangRailsback.com

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

Last changed: October 10, 2011

Contents

List of Figures **6**

List of Tables **9**

1. Objectives and Overview **10**
1.1. Document Objectives 10
1.2. Overview of *inSALMO* 1.0 10

2. Terminology and Conventions **11**
2.1. Terminology 11
2.2. Conventions 13
2.2.1. Measurement units 13
2.2.2. Parameter and variable names 14
2.2.3. Survival probabilities and mortality sources 14
2.2.4. Dates 15
2.2.5. Fish ages and age classes 15
2.2.6. Habitat dimensions and distances 15
2.2.7. Logistic functions 15

3. Habitat **16**
3.1. Reaches 16
3.1.1. Reach-scale variables 16
3.1.2. Reach links 17
3.2. Cells 19
3.2.1. Cell boundaries and dimensions 19
3.2.2. Depth and velocity 20
3.2.3. Velocity shelter availability 21
3.2.4. Spawning gravel availability 21
3.2.5. Distance to hiding cover 21
3.2.6. Food production and availability 22

4. Fish **23**
4.1. Spawning 24
4.1.1. Decide when to spawn 24
4.1.2. Select spawning cell and move there; redd guarding by female spawners 26
4.1.3. Create a redd; set number of eggs 31
4.1.4. Select a male spawner 31
4.1.5. Incur weight loss 32
4.2. Habitat Selection and Outmigration 32
4.2.1. Competition for resources via dominance hierarchy 33
4.2.2. Identify potential destination cells 33
4.2.3. Evaluate potential destination cells 36
4.2.4. Move to best destination and consume its resources 38
4.2.5. Outmigration as a habitat selection alternative 38
4.2.6. Removal of outmigrants 42
4.3. Feeding and Growth 43
4.3.1. Overview 43
4.3.2. Activity budget 46
4.3.3. Food intake: drift feeding strategy 46
4.3.4. Food intake: active searching strategy 52
4.3.5. Food intake: maximum consumption 53

4.3.6. Food intake: daily food availability 55
4.3.7. Respiration costs and use of velocity shelters 55
4.3.8. Other energy losses 57
4.3.9. Feeding strategy selection, net energy benefits, and growth 57
4.4. Fish Survival 61
4.4.1. High temperature 63
4.4.2. High velocity 64
4.4.3. Stranding 69
4.4.4. Poor condition 70
4.4.5. Terrestrial predation 71
4.4.6. Aquatic predation 78
4.4.7. Demonic intrusion: experimenter-induced mortality 84

5. Redds 85
5.1. Survival 85
5.1.1. Dewatering 86
5.1.2. Scouring and deposition 86
5.1.3. Low temperature 89
5.1.4. High temperature 90
5.1.5. Superimposition 91
5.2. Development 93
5.3. Emergence 93
5.3.1. Emergence timing 94
5.3.2. New fish attributes 94
5.3.3. Superindividual subadults 95
5.4. Empty Redds 98

6. Initialization 98
6.1. Habitat Initialization 98
6.2. Fish Initialization 98
6.2.1. Spawner arrival dates 99
6.2.2. Initialization of spawner numbers, locations, and variables 99
6.3. Redd Initialization 100

7. Random Number Generation 100

8. Scheduling 100
8.1. Habitat Update Actions 101
8.2. Fish Actions 101
8.3. Redd Actions 102
8.4. Observer Actions 102
8.5. Complete Schedule 102

9. Literature Cited 104

List of Figures

Figure 1. Representations of space in *inSALMO* 1.0. Top: polygonal cells generated in a geographic information system (GIS) or two-dimensional hydrodynamic model. Bottom: rectangular cells from a pseudo-two-dimensional hydraulic model. These are each a plan (top-down) view of one reach. 11

Figure 2. Example reach network configurations, showing junction numbers and reach names. Arrows represent reaches, pointing in the downstream direction. Network A has four sequential reaches generated by using two copies each of an upper and lower study site. Network B has two mainstem reaches and a tributary. Network C has reaches on either side of an island. 18

Figure 3. Example depth and velocity input for a cell. Each point represents an entry in the water surface elevation and velocity lookup table that is input for the cell. 20

Figure 4. Probability that an adult has spawned vs. how many days it has been in the model, for values of *fishSpawnProb* from 0.1 to 0.4. 25

Figure 5. Spawning suitability function for depth from Table 2. 29

Figure 6. Spawning suitability function for velocity, with values from Table 3. 30

Figure 7. The maximum distance fish can move, as a function of their length, for *fishMoveDistParamA* = 50, *fishMoveDistParamB* = 2. 34

Figure 8. The outmigration success function of fish length, with *fishOutmigrateSuccessL1* and *fishOutmigrateSuccessL9* set to 5.0 and 10.0 cm. 40

Figure 9. Results of the outmigration decision method with *fishOutmigrateSuccessL1* equal to 5.0 and *fishOutmigrateSuccessL9* at 10.0 cm. Light grey regions indicate combinations of growth and risk conditions under which juvenile salmon migrate downstream, and dark grey regions indicate where salmon remain in the current reach. The X axis is the daily probability of surviving factors other than starvation (e.g., predation, high temperature). The Y axis is daily growth rate as grams of growth per gram of fish weight. The four panels show how results depend on fish length. Fish were assumed to currently be in good condition; lower condition (weight at length) slightly increases the tendency to migrate downstream. 42

Figure 10. Relation between fish length and prey detection distance observed by Schmidt and O'Brien (1982), for arctic grayling feeding on 0.2 cm zooplankton. 48

Figure 11. Relative detection distance vs. turbidity: model and data of Sweka and Hartman (2001) used to fit the model. 49

Figure 12. Capture success model and the laboratory observations it was based on. 51

Figure 13. Temperature function for *cMax*. 54

Figure 14. Variation in food intake with velocity for two sizes of salmonid, using drift or search feeding. Intake is depicted as percent of *cMax* (physiological maximum daily intake). 60

Figure 15. Variation in growth rate with velocity for two sizes of salmonid, drift and search feeding strategies. Growth is depicted as percent of body mass per day. 60

Figure 16. Variation in growth rate with velocity, under winter conditions. 61

Figure 17. Survival probability function for high temperature. Daily survival (solid line) is the probability of a salmon surviving high temperature mortality for one day. 30-d survival (dashed line) is the probability for surviving the temperature for 30 days (equal to the daily survival raised to the power 30). 64

Figure 18. Maximum sustainable swimming speed as a function of fish length; measurements made at 10-15º C. The points marked as open squares were omitted as outliers. 66

Figure 19. Variation in maximum sustainable swim speed with temperature. Observations from four studies are shown separately. The Y axis is the measured swim speed divided by the speed measured at (or near) 15º in the same study. 67

Figure 20. Survival probability function for high velocity. The X axis is the fish's actual swimming speed divided by its maximum sustainable swimming speed. 68

Figure 21. Survival probability function for stranding, showing the probability for surviving one day (solid line) and for 30 days (dashed line). 69

Figure 22. Survival probability function for poor condition. The dotted line is the probability for surviving for 30 d at the value of *K*. 71

Figure 23. Depth survival increase function for terrestrial predation survival. 73

Figure 24. Turbidity survival function for terrestrial predation survival. 74

Figure 25. Fish length survival increase function for terrestrial predation survival. 75

Figure 26. Feeding time function for terrestrial predation mortality. 75

Figure 27. Velocity survival increase function for terrestrial predation survival, with parameters for both small streams and large rivers. 76

Figure 28. Distance to hiding cover function for terrestrial predation survival. 77

Figure 29. Hiding cover survival increase function for aquatic predation. 79

Figure 30. Depth survival increase function for aquatic predation survival. 80

Figure 31. Fish length survival increase function for aquatic predation survival. 81

Figure 32. Temperature survival increase function for aquatic predation survival. 82

Figure 33. Turbidity survival increase function for aquatic predation survival. 83

Figure 34. Example redd scour and fill survival function. The Y axis is the probability of a redd being destroyed during a peak flow event. 88

Figure 35. Low temperature redd survival function, for fall Chinook salmon parameter values..90

Figure 36. High temperature redd survival function, for Chinook salmon parameter values.....91

Figure 37. Redd development time (days from redd creation to emergence) vs. temperature for chinook salmon; the graph depicts the denominator of the equation for *reddDailyDevel*. ...93

Figure 38. Effect of *juvenileSuperindividualRatio* on juvenile abundance. Results are means of 50 replicates. Panel b is simply a zoom in on March 1 to July 1 results.....97

Figure 39. Effect of *juvenileSuperindividualRatio* on juvenile mean length. Results are means of 50 replicates. 98

Figure 40. Example distributions of spawner arrivals (number of arrivals in 5-day periods, out of 1000 total adults) for four values of *fishArrivalRatio* (0.01–0.5), with *fishArrivalStartDate* and *fishArrivalEndDate* set to 8/26/2005 (Julian day 238) and 11/25/2005 (Julian day 329). 99

Figure 41. Complete schedule of daily actions. 103

List of Tables

Table 1. Junction numbering for the example reach networks. 19

Table 2. Spawning depth suitability parameters, with example values for Chinook salmon developed from information in USFWS (2011). The value of *fishSpawnDSuitD1* is a depth; the value of *fishSpawnDSuitS1* is the corresponding suitability value; *fishSpawnDSuitS2* is the suitability for the depth specified by *fishSpawnDSuitD2*, etc. 29

Table 3. Spawning velocity suitability parameters, with values for Chinook salmon developed from information in USFWS (2011). The value of *fishSpawnVSuitS1* is the suitability corresponding to the velocity specified by *fishSpawnVSuitV1*, etc. 30

Table 4. Example parameter values for fish movement distance. 34

Table 5. Example parameter values for the length-weight relation, for length in cm and weight in g. 46

Table 6. Detection distance and capture success parameters. 52

Table 7. Parameter values for allometric function of maximum consumption. 54

Table 8. Parameter values for temperature function of maximum consumption. Each row in the table defines one of the points in Figure 13. 55

Table 9. Parameter values for respiration. 57

Table 10. Parameter values for high temperature mortality. 64

Table 11. Parameter values for high velocity mortality. 68

Table 12. Parameter values for stranding mortality. 70

Table 13. Parameter values for poor condition mortality. 71

Table 14. Parameter values for terrestrial predation mortality. Values are estimates for mid-sized spawning streams and should be re-considered for each site. 78

Table 15. Parameter values for aquatic predation mortality. Parameter values are for sites where large-gaped, warm- and coldwater predators are abundant. 84

Table 16. Parameter values for low temperature redd mortality. 89

Table 17. Parameter values for high temperature redd mortality. 91

Table 18. Parameter values for size of newly emerged fish. 95

1. Objectives and Overview

1.1. Document Objectives

This document describes version 1.0 of *inSALMO*, an individual-based salmon population model. Version 1.0 of *inSALMO* is focused on the effects of reservoir and river management alternatives on freshwater life stages: spawning, redd incubation, and juvenile rearing and outmigration. It was developed by Lang, Railsback and Associates (LRA) and USDA Forest Service, Pacific Southwest Research Station, for the US Bureau of Reclamation.

This document is prepared primarily as a model description help file to be packaged with the *inSALMO* software. It contains a complete description of the model's assumptions, methods, and parameters; and their basis. Additional information on the background, history, application, and software of *inSALMO* is available in the project report (Railsback et al. 2011), to which this is an appendix.

1.2. Overview of *inSALMO* 1.0

inSALMO 1.0 represents three life stages of salmon and the stream habitat they occupy. This section very briefly summarizes the kinds of things in the model and the actions they execute. Complete detail is provided in later sections.

Habitat is represented as one or more "reaches" of stream spawning and rearing habitat. Each reach is made up of a collection of polygonal cells. Flow, temperature, and turbidity are characteristics of reaches, whereas depth and velocity vary among cells. Cells can be either irregular or rectangular, depending on what kind of hydraulic model is used (Figure 1). There are no restrictions on how many cells can be in a reach, how many reaches can be in a model, or how multiple reaches are arranged spatially.

The three salmon life stages in the model are spawners, redds, and pre-smolt juveniles. Model runs start with adult salmon arriving in the reaches ready to spawn. Female spawners select a spawning cell and create a redd, then defend the redd from superimposition until they die. Redds are simulated as individual objects, with the number of live eggs in each possibly being decreased by a variety of mortality sources. Redd development status is a function of water temperature. When redds are fully developed, one new juvenile is created for each surviving egg. The model can simulate multiple races and species of salmon, though its primary application has been to fall-run Chinook.

Juvenile salmon have three behaviors: selecting habitat, feeding and growing, and surviving or dying. They select habitat cells to provide a good tradeoff between growth and survival of predation. If none of the cells they can choose among provide a good tradeoff, they can migrate downstream; the rate of downstream migration increases with juvenile size. The model's primary result is the number, timing, and size of juveniles at the time they migrate downstream out of the simulated habitat.

The model runs at a one-day time step. Key inputs describe the shape and cover characteristics of the reachs, and daily flow, temperature, and turbidity. Hence, the model is suited for predicting how changes in instream flow and temperature, and habitat restoration projects, affect spawning success.

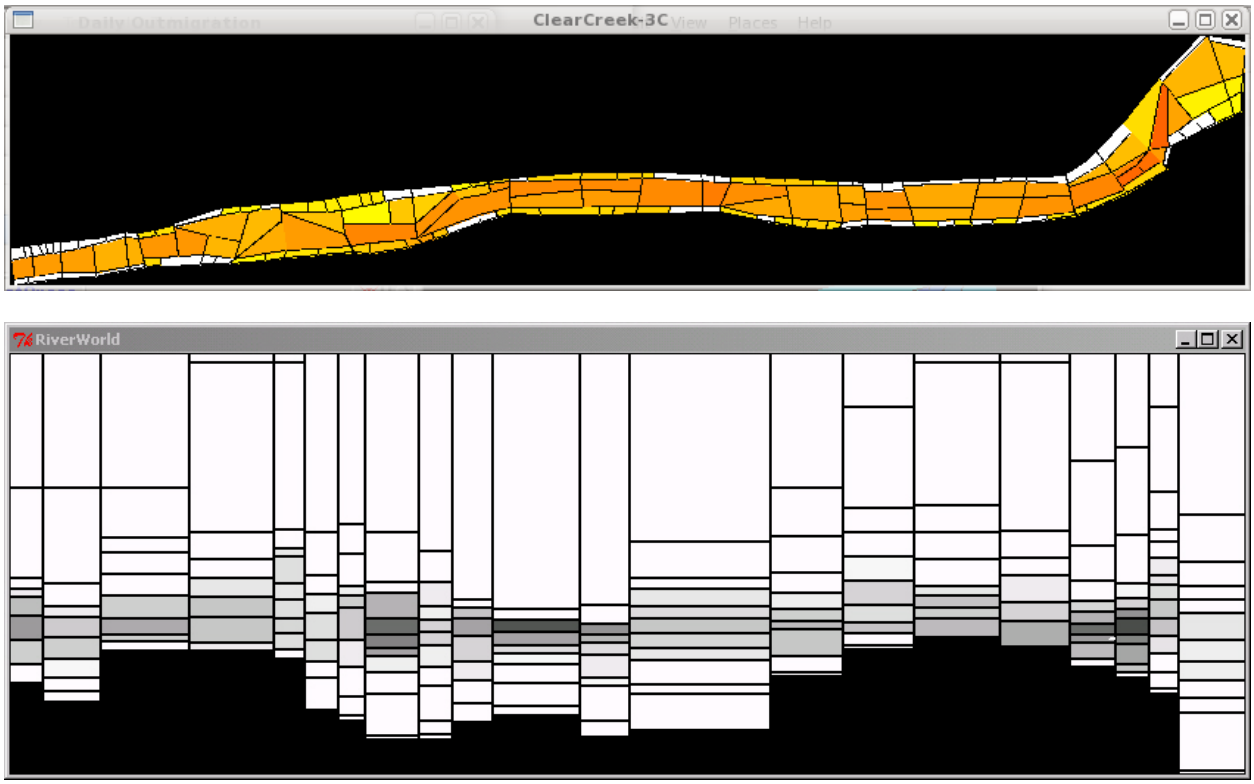


Figure 1. Representations of space in *inSALMO* 1.0. Top: polygonal cells generated in a geographic information system (GIS) or two-dimensional hydrodynamic model. Bottom: rectangular cells from a pseudo-two-dimensional hydraulic model. These are each a plan (top-down) view of one reach.

2. Terminology and Conventions

This section describes the terms and modeling conventions followed in this document and in the *inSALMO* software.

2.1. Terminology

The following terms are used as defined here throughout this document. Much of the terminology is taken from Grimm and Railsback (2005).

Action. An element in an IBM's **schedule**. An action is defined by a list of model **objects**, the **methods** of these objects executed by the action (e.g., **traits** of fish, updating the habitat cells; producing output), and the order in which the objects are processed.

Adult. Simulated salmon that have returned from the ocean to their spawning areas ready to spawn. **Spawners** are synonymous with adults.

Behavior, individual behavior, system behavior. What a model fish or fish population actually does during a simulation. A behavior is an outcome of an IBM and the ***traits*** of its individuals.

Cell. The basic unit of habitat in *inSALMO*; habitat conditions vary among cells, but not within a cell.

Data. Input that describes the habitat and fish population to be simulated. Data for *inSALMO* Version 4 includes daily time series of flow, temperature, and turbidity; cell dimensions and state variables; the relations between flow and depth and velocity for each cell; and the characteristics of the initial fish population.

Habitat selection. The **behavior** and corresponding **trait** for selecting which **cell** to feed in each day.

Input. Any of the **data** and **parameter** values that a user provides to *inSALMO* to define a **scenario**.

Method. In object-oriented software, a block of code that executes one particular **trait** or process. Methods are similar to subroutines in non-object-oriented software.

Mortality source. A natural process (e.g., starvation, predation) that causes fish or eggs to die. Mortality sources are modeled as **survival probabilities**.

Object. Something that is represented as a discrete entity with its own state variables. Example objects include individual fish, redds, and cells; and (in the software) observer tools such as graphics windows and the devices that produce output files.

Observation, observer tools. The process of collecting data and information from the IBM; typical observations include graphical display of patterns 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.

Parameter. A user-specified coefficient for one of the equations used to define traits of fish and habitat. Parameter values are one of several kinds of **input**. Compare to **variable**. Parameter values are ideally developed from empirical literature or field data. A few parameters are best evaluated via calibration.

Population. All the model fish in a simulation. (Or, for simulations with multiple species, all the model fish of a species.)

Reach. *inSALMO* models salmon spawning in one or several reaches. Each reach is a continuous section of a stream or river channel. The habitat within a reach is broken into **cells**.

Replicates. Multiple models runs that represent the same **scenario** but use different pseudo-random number sequences. Replicates are useful for evaluating how much of the variation in results is due to stochasticity.

Scenario. A single, complete set of **input** to *inSALMO*, representing one particular set of environmental conditions or one management alternative. Effects of alternative environmental conditions or management alternatives are typically assessed by comparing output produced by several different scenarios.

Schedule. A description of the order in which events are assumed to occur: 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 when they are executed.

Spawner. An **adult**.

Survival probability. A model of a **mortality source**. This term refers to a fish's probability of surviving a particular kind of mortality for one day; but it also refers to the methods used to calculate that probability.

State, state variable. A measure of the status of some part of a model (individuals, habitat cells, the population) that typically can be described using a single number. A state variable is a model variable describing a particular state of some model component. State variables may be constant over time and read from input **data**, or may be updated over time by model calculations. Example fish states are weight, sex, and location; cell state variables include distance to hiding cover (a constant input) and food availability (which varies daily); example system states are population biomass, number of species, and mortality rate (number of individuals dying per time step).

Submodel. A part of an IBM's **formulation** that represents one **trait** or process. Dividing *inSALMO* into submodels allows each process to be modeled, calibrated, and tested separately.

Trait. A model of a particular **behavior** of individual fish. A trait is a set of rules for what individuals do at particular times or in response to specific situations in the IBM.

Variable. Any number used in calculations. A variable may be a **parameter** or a **state variable**, or may be a temporary internal variable.

2.2. Conventions

2.2.1. Measurement units

The *inSALMO* 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 are in cm per second (cm/s). There are two important 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 measure of stream flow. Habitat input files that define the size and location of cells use distances in meters (m) for convenience. However, all internal variables and outputs involving depth, velocity, area, or distance use length units of cm.

Weight is in grams (g).

Temperature is in Centigrade (°C).

Turbidity is in nephelometric turbidity units (NTU).

Time is in days (d), because the model uses a daily time step. However, there are several exceptions to this convention. Flow and velocity variables are per second. Food availability and intake calculations use hourly rates because the number of hours per day that fish feed is variable.

Fish lengths are fork lengths.

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

2.2.2. Parameter and variable names

The model's formulation uses the parameter and variable naming conventions of the Swarm software used to code the model. This convention has two benefits. First, the variable and parameter names in the formulation document can be the same as in the software. Second, the names are long and descriptive, making it easier to identify exactly what each variable is.

Variable and parameter names typically are made by joining several words. The first word starts with a lower-case letter, and capital letters are used at the start of each subsequent word (e.g., *fishWeightParamA*). Input parameter names start with the kind of object that uses the parameter. These objects include fish, redds, habitat cells, fish mortality sources, and redd mortality sources. Consequently, most parameters start with the words *fish*, *redd*, *cell*, *hab*, *mortFish*, or *mortRedd*. This convention is not strictly followed for variables calculated internally by the model.

Whereas the traditional way of depicting a fish's length-weight relationship is:

$$W = a_L L^{b_L},$$

the same relationship appears in this formulation as:

$$fishWeight = fishWeightParamA \times \left(fishLength \right)^{fishWeightParamB}$$

and the corresponding program statement in the software is:

```
fishWeight = fishWeightParamA * pow(fishLength, fishWeightParamB);
```

2.2.3. Survival probabilities and mortality sources

A number of factors can cause fish or fish eggs to die in *inSALMO*. These factors are referred to as "mortality sources". Although the word "mortality" is used in parameter names and our text, all mortality-related calculations are based on survival probabilities. A survival probability is the (unitless) probability of surviving a particular 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 the individual survival probabilities together. Likewise, the probability of surviving one kind of mortality for *n* days can be calculated by raising the daily survival probability to the power *n*.

2.2.4. Dates

This model uses date input in the "MM/DD/YYYY" format (e.g.: 12/07/1999). The software converts this input to the computer operating system's internal date format that automatically accounts for leap years. All input data and simulations, therefore, include leap days.

Parameters that are days of the year (e.g., spawning is allowed to occur between April 1 and May 31 of each year) are input in the "MM/DD" day format.

2.2.5. Fish ages and age classes

inSALMO uses the convention that fish are age 0 when born and the age of all fish is incremented each January 1. (However, if a simulation starts on January 1 the birthday is skipped.) Fish are assigned to age classes, which are used to define the initial population at the start of a model run and to report simulation results. However, adult age has no effect in version 1.0 of *inSALMO* so, by convention, all adults are simply given an age of 5.

Seven age classes are used (although the number of classes can be changed via relatively simple modifications to the software):

- Age 0—fish that have not yet reached their first January 1.
- Age 1—fish that have survived one January 1.
- Age 2-Age5—fish that have survived the January 1 of two (etc.) years.
- Age 6+—any fish older than Age 5.

2.2.6. Habitat dimensions and distances

X and Y dimensions. Version 1.0 of *inSALMO* uses a general two-dimensional format for depicting space. Cells can have 3 or more sides. Velocities are modeled only as magnitudes without any direction. The X and Y coordinates used to define cell corners can be in any rectangular coordinate system. On *inSALMO*'s graphical displays of the stream, the X coordinate is the horizontal dimension and increases from left to right; Y is the vertical dimension and increases from bottom to top. Hence, input in standard coordinate systems such as UTM appear with north on the top and east to the right.

Distances between cells. Some calculations in the model require values for the distance between two cells (e.g., for finding all the cells within a fish's maximum movement distance). The distance between two cells is calculated as the straight-line distance between the centroids of the cells.

2.2.7. Logistic functions

The survival probabilities make extensive use of logistic functions, which are useful for depicting many functions that vary between 0 and 1 in a nonlinear way. The Y value of a logistic function

increases from zero to one, or decreases from one to zero, as the X value increases over any range. In *inSALMO*, logistic functions are defined via parameters that specify two points: the X values at which the Y value equals 0.1 and 0.9. The logistic functions are defined as:

$$S = \exp(Z) \big/ 1 + \exp(Z)$$

where

$$Z = LogistA + (LogistB \times habitatVariable)$$

$$LogistA = \left(LogistC - LogistD\right) \big/ \left(habVarAtS01 - habVarAtS09\right)$$

$$LogistB = LogistC - \left(LogistA \times habVarAtS01\right)$$

$$LogistC = \ln\left(0.1 \big/ 0.9\right)$$

$$LogistD = \ln\left(0.9 \big/ 0.1\right).$$

These equations evaluate the example survival probability *S*, given the X value *habitatVariable*. The parameters *habVarAtS01* and *habVarAtS09* are the values of the habitat variable at which survival is defined to be 0.1 and 0.9, respectively. The two X value parameters (*habVarAtS01* and *habVarAtS09* in this example) must not be equal. (Many examples of logistic functions are shown graphically in Section 4.4.)

3. Habitat

Habitat is depicted in *inSALMO* at three scales. The entire model is represented as a network of reaches (but just one reach can be used). Reaches are habitat objects representing a whole river or stream segment, and cells are objects representing the habitat units that salmon occupy. A model contains one or more reaches, and each reach is made up of many cells.

3.1. Reaches

Reaches represent variables and processes that are assumed uniform over a reach. Reaches also keep track of how they are linked to other reaches.

3.1.1. Reach-scale variables

The parameters used to calculate food production in each cell (Section 3.2.6) are assumed uniform over a reach and constant over time. Hence, they are input as reach parameters. Two other reach-level parameters are the maximum flow at which salmon will spawn (Section 4.1.1) and the fraction by which velocities are reduced for salmon swimming in velocity shelters (Section 4.3.7).

Reaches have three variables that are updated daily from input files: daily mean values of flow (m^3/s), temperature (˚C), and turbidity (NTU). Temperature and turbidity are assumed the same

for all cells in a reach. Flow is used primarily to determine the depth and velocity in each of the reach's cells (Section 3.2.2).

Flow is also used to calculate the probability of redd scour (Section 5.1.2). Scour is actually a function of peak flow, and peak flows during high-flow events can be significantly higher than daily mean flows. Hence, during peak flow periods that coincide with redd incubation, *inSALMO* may better represent redd scour if daily mean flows are replaced by daily peak flows. This change should have negligible other effects on model results because (unless multiple salmon races are simulated) few fish are present during the incubation period. (Any fish that are present could be affected by the higher velocities resulting from peak vs. mean daily flows.)

The day length (*dayLength*, number of hours of daylight, including twilight) is a calculated reach variable. (The same day length is used for all reaches.) Day length is used to model the time salmon spend feeding (Section 4.3.2) and affects predation mortality (Section 4.4). The value of dayLength is updated daily, using equations modified from the Qual2E water quality model (Brown and Barnwell 1987).

$$dayLength = 24 - 2\left[\left(\frac{12}{\pi}\right)\arccos\left\{\tan\left(\frac{\pi \times siteLatitude}{180}\right)\tan\delta\right\}\right]$$

where:

$$\delta = \left[\left(\frac{23.45}{180}\right)\pi\cos\left\{\left(\frac{2\pi}{365}\right)\left(173 - julianDate\right)\right\}\right]$$

and *siteLatitude* is a model parameter set to the study site's latitude (in degrees) and *julianDate* is the Julian date (day of the year, 1-366, calculated internally from the date). This equation works only for the northern hemisphere.

3.1.2. Reach links

Users of *inSALMO* specify the number of reaches and how they are linked. (Often, only one reach is used.) Reaches can be linked in a network of any kind, including a linear sequence (multiple mainstem reaches only), mainstem and tributaries, and distributaries (Figure 2).

The reach network is specified by providing, for each reach, a reach name and junction numbers for the upstream and downstream ends of the reach. The reach name is a character string of up to 30 characters with no spaces. The reach name is used within the software and in output files to label each reach.

For each reach, junction numbers are provided as two reach parameters: *habUpstreamJunctionNumber* and *habDownstreamJunctionNumber*; both are integers. Junction numbers are used only to build the links that define the reach network, so their value can be arbitrary as long as they are consistent among reaches. Any two or more reaches with the same junction number will be linked at that junction. Figure 2 illustrates ways that networks of reaches can be defined, and Table 1 describes how these networks are defined using junction numbers.

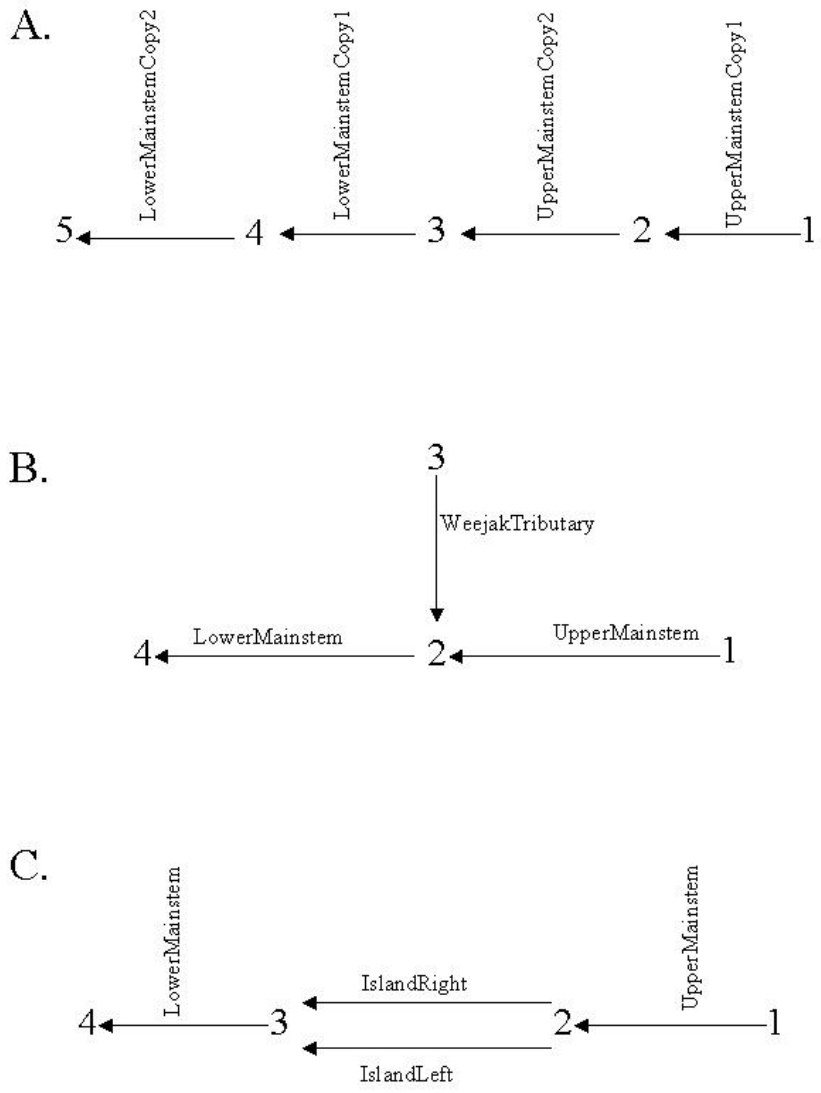


Figure 2. Example reach network configurations, showing junction numbers and reach names. Arrows represent reaches, pointing in the downstream direction. Network A has four sequential reaches generated by using two copies each of an upper and lower study site. Network B has two mainstem reaches and a tributary. Network C has reaches on either side of an island.

Table 1. Junction numbering for the example reach networks.

Network	Reach name	Upstream junction number	Downstream junction number
A	UpperMainstemCopy1	1	2
	UpperMainstemCopy2	2	3
	LowerMainstemCopy1	3	4
	LowerMainstemCopy2	4	5
B	UpperMainstem	1	2
	LowerMainstem	2	4
	WeejakTributary	3	2
C	UpperMainstem	1	2
	IslandLeft	2	3
	IslandRight	2	3
	LowerMainstem	3	4

3.2. Cells

3.2.1. Cell boundaries and dimensions

Cells are depicted as polygons with three or more sides. A reach's cells can be laid out using GIS software, or as the mesh of a hydraulic model—either a pseudo-two-dimensional model such as those of PHABSIM or a fully two-dimensional hydrodynamic model. *inSALMO* imports the corner coordinates of each cell. Any space not within a cell is treated as unavailable to the model fish.

Cells also have a variable *cellReachEnd* that simply holds a static value indicating whether the cell is at the upstream end (*cellReachEnd* = "U") or downstream end ("D") of their reach, or not at an end ("I"). These variables are used by fish in determining which cells of which reaches they potentially could move to (Section 4.2.2.2). The values of *cellReachEnd* are determined by the model user and read in as input. Cells should have *cellReachEnd* equal to "U" or "D" only if they are at the end of their reach *and* are wet at typical base flows.

3.2.2. Depth and velocity

The depth and velocity of each cell (and the number of cells that are submerged and therefore available to salmon) vary with the daily reach flow. A cell's water velocity is treated as a magnitude: the mean speed of the water without regard to direction.

To take advantage of existing stream hydraulic modeling software and avoid having to include hydraulic simulations, *inSALMO* imports lookup tables of water depth and velocity, as a function of flow, for each cell. This approach allows all the hydraulic model building, testing, and calibration to be conducted in existing, specialized hydraulic software and manipulated, if necessary, in GIS.

The input depth and velocity lookup tables should contain a wide range of flows. If *inSALMO* 4.5 is used to simulate flows higher than those in the lookup table input it is likely to produce unrealistic depths and velocities for some cells.

An example is depicted graphically in Figure 3. In this example, the cell is dry (depth and velocity are zero) at flows up to 20 m^3/s. As flow increases, depth increases steadily. Velocity at this example cell, however, does not increase monotonically with flow: it increases rapidly with flows between 25 and 30, then drops off, then increases sharply at flows around 85. Such discontinuities in how velocity increases with flow in part an artifact of how the hydraulic simulations were done (three hydraulic model calibrations were used for low, middle, and high ranges of flow) but also reflect the discontinuities that really occur in rivers. Because of eddies and other hydraulic complexities, it is not unusual for velocity to decrease in a cell as flow increases over some ranges. (This example is slightly atypical: velocity does increase monotonically with flow at most cells. However, exceptions like this are common; the example is presented to reinforce that capturing natural hydraulic complexity should be the highest priority in hydraulic simulation.)

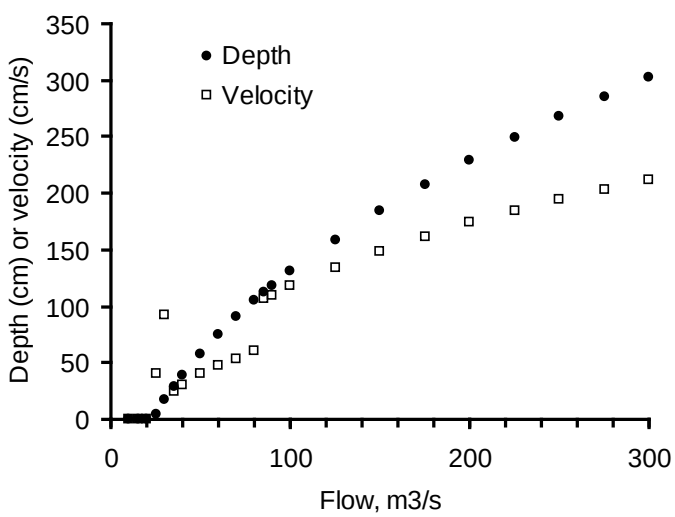


Figure 3. Example depth and velocity input for a cell. Each point represents an entry in the water surface elevation and velocity lookup table that is input for the cell.

On each simulation day, the depth and velocity of each cell are interpolated from the reach's daily flow, using the lookup tables. Linear interpolation is used, so it is important for the lookup table to include many flows. For flows above the highest in the lookup table, depth and velocity are extrapolated upward from the highest two flows in the table. The need to make these extrapolations can be avoided by making sure the lookup table includes flows lower and higher than any occurring during a model run.

At flows below the lowest in the lookup table, depth is extrapolated downwards from the lowest two values in the table; any negative results are set to zero depth. Velocity is interpolated in this case between zero and the velocity at the lowest flow in the table. Any channel margin cells that are submerged only at the highest flow in the lookup table can have unrealistically high velocities at flows above the highest lookup table flow. Cells that are submerged only at flows above the highest lookup table flows will never have non-zero depth and velocity (because all values in their lookup table are zero).

3.2.3. Velocity shelter availability

The availability of velocity shelters (which affect growth, Section 4.3.7; and high velocity mortality, Section 4.4.2) is modeled by assuming that a constant (over time) fraction of each cell's area provides velocity shelter. This fraction is provided as input (variable *cellFracShelter*, a dimensionless fraction between zero and one). These fractions should include any part of the cell with complex hydraulics that could be used by salmon to reduce their swimming speed while drift feeding. Velocity shelters can be provided by boulders, cobbles or other substrates that induce roughness in the bottom, woody debris, roughness in the banks or bedrock channel, or adjacent cells with near-zero velocities.

In reality, the availability of velocity shelters can vary with a fish's size and the flow; *inSALMO* ignores this variability because of its complexity. Instead, *cellFracShelter* should represent drift feeding habitat for juvenile salmon at a typical base flow.

A cell keeps track of its total velocity shelter area (*cellFracShelter* × *cellArea*) and also keeps track, over time, of how much of that shelter area is occupied by fish. Each fish using velocity shelter in a cell occupies an area of shelter equal to the square of the fish's length (Section 4.3.7). A fish has access to shelter if the total shelter area of its cell is greater than the shelter area already occupied by more dominant fish. This means that a fish has access to shelter if there is <u>any</u> unused shelter space available for it in the cell. (Competition for food, not velocity shelter space, is more likely to limit the density of fish in a cell.)

3.2.4. Spawning gravel availability

Spawning gravel availability is described as the fraction of cell area with gravel suitable for salmon spawning, assumed to be constant over time. This spawning gravel fraction (variable *cellFracSpawn*, a dimensionless fraction between zero and one) is provided as input for each cell.

3.2.5. Distance to hiding cover

The habitat input variable *cellDistToHide* (m) is an estimate of how far a fish in the cell would have to move to find hiding cover. This variable is used in the terrestrial predation mortality model (Section 4.4.5). The kind of habitat that salmon can use for hiding varies with fish size.

Because the focus of *inSALMO* is on juvenile rearing, the terrestrial predation formulation is designed so that *cellDistToHide* should represent hiding for juveniles.

3.2.6. Food production and availability

The amount of food available to fish is a very important habitat variable, probably more important than flow or temperature in determining fish population abundance and production except under extreme conditions. Unfortunately, the processes influencing food availability for stream salmonids are complex and not well understood. Although some studies (Gowan and Fausch 2002, Morin and Dumont 1994, Railsback and Rose 1999) indicate that food availability and consumption can vary with factors including flow, temperature, fish abundance, and physical habitat characteristics, there is little information available on how food availability varies over time and space at scales relevant to individual-based models. Modeling food production is also complicated by the multiple sources of food available to fish. Stream salmonids are commonly observed feeding both by "drift feeding"–maintaining a stationary position and capturing food that drifts past; and by "search feeding"–actively searching for food on the stream bottom or surface. *inSALMO* separately models "drift" food that moves with the current and "search" food that is relatively stationary and must be searched out by the fish. Both drift and search food may originate with benthic production or from terrestrial input.

Because *inSALMO* assumes fish compete for the food available in each cell, cells must keep track of: (a) how much food of each type is produced each day; and (b) how much is available to a particular fish.

3.2.6.1. ***Production***

In the absence of established models of salmon food availability, *inSALMO* uses models that are simple yet mechanistic and easily calibrated using observed salmon growth and survival. Food production is modeled using the simple assumption that (1) the concentration of food items in the drift (*habDriftConc*, grams of prey food per cm^3 of stream volume) and (2) the production of search food items (*habSearchProd*, grams of prey food produced per cm^2 of stream area per hour) are constant over time and space. These two variables are input as habitat parameters.

[How food is produced in specific habitats such as riffles, and depleted by fish as it travels downstream, has been simulated in other models (e.g., Hughes 1992a). However, the model of Hughes (1992a) shows that simulating drift production and depletion over space would require a major increase in the complexity. The simpler approach used in *inSALMO* appears to generally capture the important dynamics of food competition.]

The salmon feeding formulation uses hourly food production and consumption rates because the number of feeding hours per day varies. The hourly food production rates are determined by the physical characteristics of habitat cells. The rate at which search food is produced in a cell (*searchHourlyCellTotal*, g/h) is simply the cell area multiplied by *habSearchProd*.

The rate at which drift food is produced in a cell (*driftHourlyCellTotal*, g/h) is modeled as the rate at which prey items flow into the cell from upstream, plus the rate at which consumed prey are regenerated within the cell:

$$driftHourlyCellTotal = 3600 \times cellArea \times cellDepth \times cellVelocity \times habDriftConc / habDriftRegenDist.$$

The constant 3600 converts the rate from per second to per hour. The *habDriftRegenDist* term has two purposes. First, it simulates the regeneration of prey consumed by drift-feeding fish. Second, it makes the amount of drift food available per cell area independent of the cell's shape. Without this term, five cells 2 m in length (in the upstream-downstream direction) would have five times the food availability of one 10 m-long cell.

The parameter *habDriftRegenDist* (cm) should theoretically have a value approximating the distance over which drift depleted by foraging fish is regenerated. Smaller values of *habDriftRegenDist* provide higher production of food in a cell. This parameter can be used to calibrate habitat selection and survival of starvation; varying it changes drift food availability without changing the amount that a drift-feeding fish captures. The parameter *habDriftConc* also affects the amount of food in a cell, but unlike *habDriftRegenDist*, also affects food capture rates of drift-feeding fish (Section 4.3.3).

3.2.6.2. ***Availability***

The amount of food available to a particular salmon affects the salmon's habitat selection and growth methods (Section 4.2.1). Food availability to a fish is modeled as the hourly rate at which food is produced but not consumed by larger fish, so is still available for other fish. Availability is tracked separately for drift and search food; these rates are *driftHourlyCellAvail* (g/h) and *searchHourlyCellAvail* (g/h). For example, a cell's drift food may be completely consumed by larger fish (*driftHourlyCellAvail* is zero) while all of its search food remains available for any fish that chooses to use search feeding (*searchHourlyCellAvail* equals *searchHourlyCellTotal*).

The cells keep track of drift and search food availability. At the start of a simulation day, *driftHourlyCellAvail* is set equal to *driftHourlyCellTotal* and *searchHourlyCellAvail* is set equal to *searchHourlyCellTotal*. As the salmon execute their habitat selection methods (Section 4.2), the rate of drift or search food consumed by any fish choosing to occupy the cell is subtracted from the food availability rate for additional fish. When a fish's consumption is limited by the amount of food available in the cell, its consumption will equal all the remaining availability and no food will be available for additional fish. Any fish moving into a cell where all the (drift or search) food is consumed by larger fish will consequently have zero (drift or search) food available for it to consume.

4. Fish

This section describes the methods used by the fish objects in *inSALMO*. These fish objects represent two of the three salmon life stages in the model; the other life stage—incubating eggs and alevins—are represented by redd objects (Section 5). With some key exceptions, both adult spawners and juvenile salmon use the same methods and parameters.

Fish daily carry out four sets of actions: spawn, select a habitat cell, feed and grow, and survive or die according to survival probabilities that vary with habitat cell and fish characteristics. The methods used in these actions are described in this section. The schedule for fish actions—the order in which they are executed—is summarized in Section 8.2.

Some of the parameters used in fish methods are clearly species-specific or site-specific. Example values for these parameters are provided here, along with information on the species, races, or sites for which they were developed. Many parameter values, however, can be considered acceptable for stream salmonids in general: whatever variation there may be in parameter values among species is expected to be unimportant compared to other variability and uncertainty in the method the parameter is used in.

4.1. Spawning

The formulation for when and where adults spawn is guided by several ideas. First, for *inSALMO*'s purposes as a river management tool it is important to simulate when and where, at a reach and cell scale, salmon spawn. Secondly though the model's purposes do not clearly require representation of spawning decisions as complex adaptive behaviors. 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, *inSALMO*'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, *inSALMO*'s spawning methods simply force model salmon to reproduce general spawning behaviors observed in real salmon. Behaviors are included only if they appear important for simulating effects of physical habitat, flow, and temperature effects on reproduction.

Spawning simulations include five steps: females decide whether to spawn, select a cell to spawn in, create a redd, and identify a male mate; then, both females and males incur a weight loss.

4.1.1. Decide when to spawn

Each day, each female spawner salmon determines whether it meets all of the fish- and habitat-based spawning criteria described below. These spawning criteria restrict spawning to physical conditions (dates, flows, temperatures) when spawning has been observed in real salmon, presumably because spawning is more likely to be successful during those conditions. The criteria for readiness to spawn do not include a requirement that good spawning habitat be available; it is assumed that salmon will spawn whether or not ideal gravel spawning habitat is present, because model input should create spawners only in reaches where they are known to spawn (Section 6.2).

On the days when all the spawning criteria are met for a female, then whether it actually spawns that day is determined stochastically. The probability of spawning on any such day is the parameter *fishSpawnProb* (unitless). This stochastic selection of spawning date imposes some variability in when individual fish spawn; the appropriate degree of variability can depend on what salmon race is being modeled and how much time typically passes between when they arrive at a site and when they spawn. Fall-run chinook typically spawn soon after arriving in their spawning reach, so *fishSpawnProb* should have a relatively high value. Values of *fishSpawnProb* of 0.1, 0.2, 0.3, and 0.4 result in half the adults spawning within 7, 4, 2, and 2 days (Figure 4).

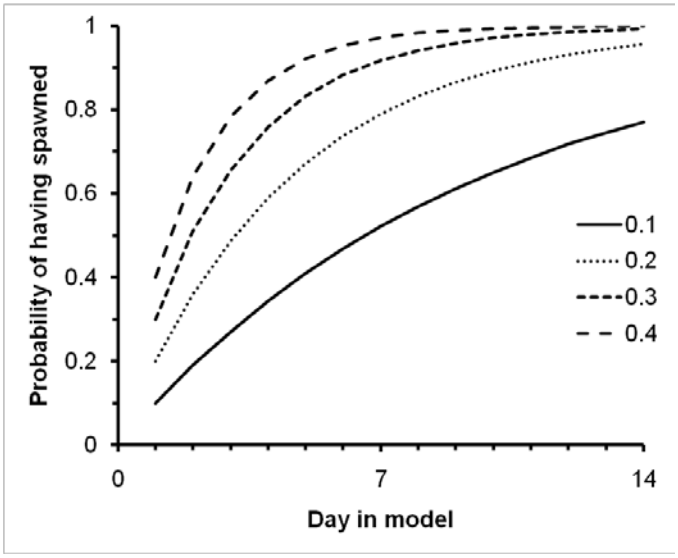


Figure 4. Probability that an adult has spawned vs. how many days it has been in the model, for values of *fishSpawnProb* from 0.1 to 0.4.

4.1.1.1. ***Not spawned this season***

Model salmon cannot spawn more than once. The fish (both males and females) in *inSALMO* have a boolean (yes-no) variable *spawnedThisSeason*. When adults are created, their value of *spawnedThisSeason* is set to NO. If a fish spawns, its value of *spawnedThisSeason* is set to YES. Females are not allowed to spawn if their value of *spawnedThisSeason* is already YES. (If a fish spawns, its value of *spawnedThisSeason* remains YES until spawning season starts again the next year.)

4.1.1.2. ***Date window***

Salmonids generally have distinct annual spawning seasons. This is not surprising because time of year is an important predictor of factors that are 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, but spawning too late may make offspring more vulnerable to high temperatures or reduce their ability to compete with earlier-spawned juveniles. Some salmon races migrate to their spawning reaches long before spawning, while others begin spawning as soon as they arrive. Therefore, in *inSALMO* fish can spawn only on days within a user-specified date window.

The date window is specified by two input parameters, *fishSpawnStartDate* and *fishSpawnEndDate*. These parameters are days in MM/DD format. (The spawning window can extend from the end of one year into the next; for example, *fishSpawnStartDate* can be 12/1 with *fishSpawnEndDate* 2/1.)

4.1.1.3. ***Temperature range***

Temperature is widely accepted as a factor controlling the timing of salmonid spawning (e.g., Lam 1988). Temperature could be used by spawners as a cue for seasonal changes and to avoid temperature-induced egg mortality. Therefore, spawning in *inSALMO* can only occur

within a range defined by parameters for maximum (*fishSpawnMaxTemp*) and minimum (*fishSpawnMinTemp*) spawning temperatures for spawning.

4.1.1.4. ***Flow limit***

The maximum flow limit implements the assumption that salmon will not spawn during high flow events. During unusually high flow, cells with depths and velocities suitable for redds (Section 4.1.2) are likely to be along river margins where redds are at risk of dewatering mortality when flows recede; and cells with good habitat for redds at normal flows are vulnerable to scouring. The high flow limit is defined by a single habitat reach parameter, *habMaxSpawnFlow* (m^3/s). A fish is not allowed to spawn if the flow in its reach is greater than *habMaxSpawnFlow*. (This is a habitat parameter instead of a fish parameter because it varies among reaches.) This parameter is highly site-specific and can only be estimated for each study site.

4.1.1.5. ***Steady flows***

Fish are assumed not to spawn when flows are unsteady because flow fluctuations place redds at risk of dewatering or scouring mortality. The parameter *fishSpawnMaxFlowChange* (unitless) is used to define this criterion: if the fractional change in flow from the previous day is greater than the value of *fishSpawnMaxFlowChange* then spawning is not allowed. This fractional change in flow is evaluated as:

$$fracFlowChange = \text{abs}(reachFlow - yesterdaysFlow)/todaysFlow$$

where *reachFlow* is the current day's flow, *yesterdaysFlow* is the flow on the previous day and abs() is the absolute value function. Van Winkle et al. (1996) and Railsback and Harvey (2001) estimated 0.20 as a reasonable value for *fishSpawnMaxFlowChange*.

4.1.1.6. ***Last spawning date***

This criterion overrides all others (including the effect of *fishSpawnProb*) and forces all spawners to spawn on the day it is true. The criterion is that adults always spawn, if they have not already, on the last day of the spawning window defined by *fishSpawnEndDate*. Normally this behavior only affects males that were not chosen as partners by any female spawners; forcing such otherwise unspawned males to spawn causes them to lose their spawning weight and therefore die (Section 4.1.5) instead of unrealistically surviving for many weeks. Females would be affected by this behavior only if environmental conditions (unsteady flows, extreme temperatures) or random chance kept them from spawning soon after arriving.

4.1.2.Select spawning cell and move there; redd guarding by female spawners

Female spawners select the cell in which they then build a redd. While selection of habitat for foraging is modeled very mechanistically (Section 4.2), 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 salmon. This decision was made because a detailed, mechanistic representation of spawning habitat selection would require considerable additional complexity: modeling processes such as intergravel flow and water quality, which are extremely data-intensive and uncertain. This additional complexity is not necessary to meet *inSALMO*'s objectives (Section 4.1), but we do need a simple representation of how flow affects where redds are placed because a redd's location affects its survival of dewatering (Section 5.1.1).

The first step in identifying the location for a new redd is identifying all the cells that are potential spawning sites. This step uses the same method used by salmon to identify potential destination cells during habitat selection (Section 4.2.2), but with two important differences.

The first difference is that a spawner only considers cells within its current reach, even if there are multiple reaches in the model. Without this rule, users would not be able to completely control how many adults spawn in each reach, which is important for testing and using *inSALMO*.

The second difference between habitat selection for spawning vs. foraging represents of redds by female spawners, which is assumed to be an important process controlling the distribution of redds and superimposition mortality for redds. This process is implemented by:

- A fish parameter, *fishSpawnDefenseArea*, represents the area (cm^2) that a female spawner excludes other spawners from as long as she lives (further explained at Section 5.1.5).

- A state variable for cells, *cellAvailableGravelArea* (cm^2) is used to track the area of undefended spawning gravel in each cell.

- Spawners base their selection of spawning cells (below) on *cellAvailableGravelArea* instead of on the cell's total area of gravel.

- During habitat updates at the start of each time step, the value of *cellAvailableGravelArea* is re-set to the cell's total area of spawning gravel minus the value of *fishSpawnDefenseArea* of each live female spawner in the cell (fish in the cell with *spawnedThisSeason* equal to TRUE and sex equal to female; remember that *fishSpawnDefenseArea* can differ among species). However, if the resulting value of *cellAvailableGravelArea* is negative, it is set to zero. This update accounts for spawners who have died and no longer defend their redd.

- Whenever a spawner creates a redd, the value of *fishSpawnDefenseArea* is subtracted from *cellAvailableGravelArea* immediately (before any other spawners select their spawning cell on the same time step). If *cellAvailableGravelArea* becomes negative, it is set to zero.

After potential spawning cells are identified, they are rated by the spawner to identify the cell where the redd will be created. The spawning cell is the potential spawning cell with the highest value of variable *spawnQuality* where:

$$spawnQuality = spawnDepthSuit \times spawnVelocitySuit \times cellAvailableGravelArea.$$

The variables *spawnDepthSuit* and *spawnVelocitySuit* are unitless habitat suitability factors determined using methods described below. (The units of *spawnQuality* are therefore cm^2, but they are unimportant.) The variable *cellAvailableGravelArea* is included in *spawnQuality* because a spawner is assumed more likely to spawn in a cell that has more area of gravel, even if it does not select for bigger patches of gravel. Superimposition redd mortality (Section 5.1.5) is likely to result from this formulation because spawners search many cells for the best spawning

habitat—so it is likely that more than one spawner will use the same cell. However, the best cell for spawning can vary from day to day as flow varies, and due to redd defense.

It is possible that none of the potential spawning cells have a value of *spawnQuality* greater than zero, especially where spawning gravel is extremely sparse. If *spawnQuality* is zero for all potential spawning cells, then the model assumes a spawner will still spawn but ignore gravel area as a criterion. In this situation, the spawner selects the cell with the highest value of *spawnQuality* ignoring spawning gravel:

$$spawnQuality = spawnDepthSuit \times spawnVelocitySuit.$$

If there are still no cells with *spawnQuality* greater than zero, then the spawner places its redd in its current cell. (This condition should occur very rarely, especially if *habMaxSpawnFlow* is well-chosen.)

When the female spawner has selected its spawning cell, the spawner moves to that cell. Male spawners are not assumed to move to the spawning cell.

The suitability factors *spawnDepthSuit* and *spawnVelocitySuit* are unitless variables representing the tendency of salmonids to select fairly well-defined ranges of depth and velocity for spawning (e.g., Knapp and Preisler 1999). Presumably, real salmon select these ranges because they correspond to hydraulic conditions under which egg survival is generally high. For example, intermediate depths have highest suitability, likely 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 *inSALMO* and available in the literature for a variety of species and sites (e.g., Gard 1997).

The spawning suitability factors for depth and velocity are interpolated linearly from suitability relations provided as parameters. Values of *spawnDepthSuit* are interpolated from the parameters in Table 2 (also plotted in Figure 5). The number of points in this suitability relationship is fixed at five.

Table 2. Spawning depth suitability parameters, with example values for Chinook salmon developed from information in USFWS (2011). The value of *fishSpawnDSuitD1* is a depth; the value of *fishSpawnDSuitS1* is the corresponding suitability value; *fishSpawnDSuitS2* is the suitability for the depth specified by *fishSpawnDSuitD2*, etc.

Parameter Name	Parameter Value (depth, cm)	Parameter Name	Parameter Value (unitless suitability)
fishSpawnDSuitD1	0	*fishSpawnDSuitS1*	0.0
fishSpawnDSuitD2	12	*fishSpawnDSuitS2*	0.0
fishSpawnDSuitD3	27	*fishSpawnDSuitS3*	0.95
fishSpawnDSuitD4	33.5	*fishSpawnDSuitS4*	1.0
fishSpawnDSuitD5	204	*fishSpawnDSuitS5*	0.0

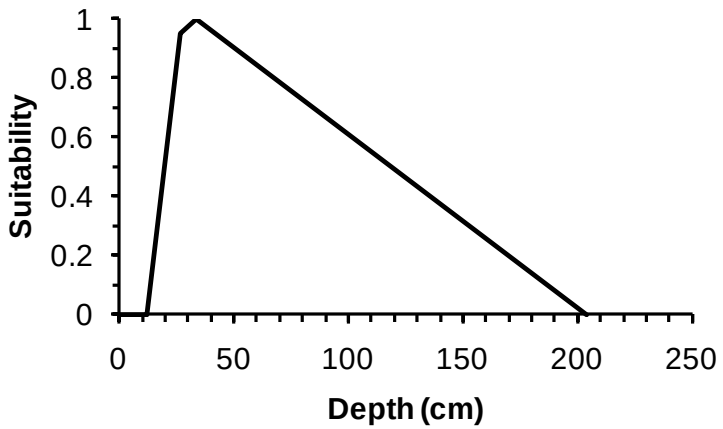


Figure 5. Spawning suitability function for depth from Table 2.

A value of *spawnVelocitySuit* for a cell is interpolated from the six pairs of parameters in Table 3, which includes example parameter values for Chinook. The number of points in this relationship is fixed at six.

Table 3. Spawning velocity suitability parameters, with values for Chinook salmon developed from information in USFWS (2011). The value of *fishSpawnVSuitS1* is the suitability corresponding to the velocity specified by *fishSpawnVSuitV1*, etc.

Parameter Name	Parameter Value (velocity, cm/s)	Parameter Name	Parameter Value (unitless suitability)
fishSpawnVSuitV1	0	*fishSpawnVSuitS1*	0.0
fishSpawnVSuitV2	2.3	*fishSpawnVSuitS2*	0.0
fishSpawnVSuitV3	3.0	*fishSpawnVSuitS3*	0.06
fishSpawnVSuitV4	54	*fishSpawnVSuitS4*	1.0
fishSpawnVSuitV5	61	*fishSpawnVSuitS5*	1.0
fishSpawnVSuitV6	192	*fishSpawnVSuitS6*	0.0

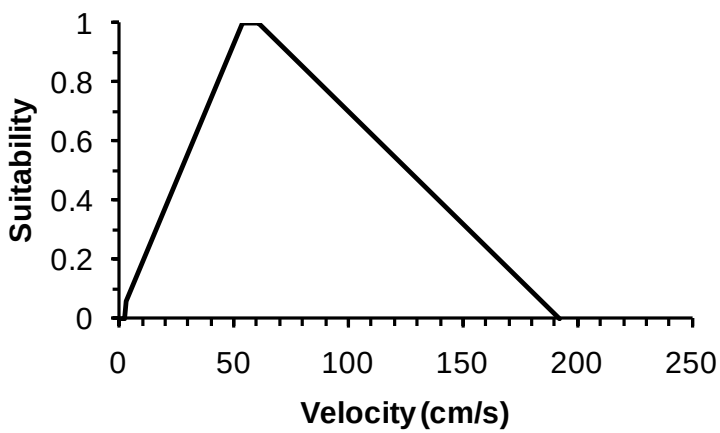


Figure 6. Spawning suitability function for velocity, with values from Table 3.

These example parameter values should be reconsidered for each site that *inSALMO* is applied to. In bigger rivers, for example, greater depths may be suitable without risk of scouring; larger spawners and greater spawning gravel size may reduce the risk of scouring, making higher velocities suitable.

If the model needs to interpolate a value of *spawnDepthSuit* for a depth greater than the value of *fishSpawnDSuitD5* (or a value of *spawnVelocitySuit* for a velocity greater than *fishSpawnVSuitV6*), the value is extrapolated from the last two points in the suitability relation. However, suitability values less than zero are converted to zero. Suitability values greater than one are allowed, so suitability could be scaled from 0 to 10 instead of 0 to 1.0. (It is actually very

unlikely that depth and velocity have exactly equal effects on redd location, so they should have different maximum suitability values.)

4.1.3. Create a redd; set number of eggs

When a female spawner has selected a spawning cell, it creates a redd in the cell. The number of eggs in the redd depends on the spawner's fecundity (a function of length) and losses during spawning:

$$numberOfEggs = \left(fishFecundParamA \times fishLength^{fishFecundParamB} \right) \times fishSpawnEggViability$$

The first term in this equation is the spawner's fecundity, the number of eggs it produces. Example values, for Sacramento River Chinook salmon (Table 1 of Healey and Heard 1984, with postorbithypural length converted to fork length) are *fishFecundParamA*: 690 and *fishFecundParamB*: 0.522.

The second term consists of the parameter *fishSpawnEggViability*, which is the fraction of eggs that are successfully fertilized and placed in the redd. (Even though *fishSpawnEggViability* has the same effect mathematically as *fishFecundParamA*, 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 not explicitly included in the model (Section 5.1). There is little published literature to support consistent values of *fishSpawnEggViability* for stream salmonids. For example, Healey (1991) reviewed egg deposition for chinook salmon and found only a few conflicting studies, concluding that egg loss could be high in high-velocity streams but is often low. Anecdotal evidence 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. A value of 0.8 seems reasonable.

4.1.4. Select a male spawner

When a female spawns, it attempts to select a male that also spawns. The only purpose of identifying a male spawner is to impose spawning weight loss (described below) on the male. The selected male spawner is the largest fish in the simulation that meets all the male spawner criteria listed below. The largest eligible 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 (e.g., for Atlantic salmon, Jones and Hutchings 2002).

This selection of a male occurs after the female creates the redd. If several females spawn on the same day, the male selected by the first female spawner becomes ineligible for the subsequent female spawners on the same day (because one of the male spawner criteria is having not previously spawned). If no male meets the criteria as a spawner, there is no effect on the female or redd. The female 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 to include in the model. Males are not assumed to move as a result of spawning.

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

- Is male;
- Is an adult spawner;
- Is of the same species as the female;
- Occupies the same reach as the female and its new redd; and
- Has not previously spawned during the current spawning season.

4.1.5. Incur weight loss

Spawning involves a significant penalty in body mass and energy, for both males and females. Simulating this loss of mass is how *inSALMO* causes adult salmon to die soon after spawning: the weight loss results in mortality due to starvation and poor condition (Section 4.4.4) soon after spawning (especially because adults are assumed not to eat; Section 4.3). When any model salmon—male or female—spawns, their weight is reduced according to the parameter *fishSpawnWtLossFraction*. Fish weight is multiplied by 1- *fishSpawnWtLossFraction*. A value of 0.4 for *fishSpawnWtLossFraction* is supported by Mesa and Magie (2006).

With the parameters for poor condition survival recommended in Section 4.4.4, a salmon losing 40% of its body mass has a daily survival probability of 0.9. This survival corresponds to a 47% probability of surviving for one week. Survival of adults after spawning affects the model results only via guarding of redds by females (Section 4.1.2).

4.2. Habitat Selection and Outmigration

Habitat selection is the most important trait of juvenile salmon in *inSALMO*. (The word “movement” is commonly also used for this trait; “habitat selection” is a more precise term but in this document the terms are generally interchangeable.) Habitat selection is very important to simulate realistically because it is probably the most important way that stream fish can adapt to short- and mid-term changes in habitat and fish state. Railsback et al. (1999) reviewed methods used in previous models and developed the approach used in *inSALMO*; the approach was then tested by Railsback and Harvey (2002).

The habitat selection trait used by salmon is conceptually simple: every day, each juvenile salmon moves to the habitat cell that (1) is close enough that the fish can be assumed to be aware of conditions in it, and (2) offers the highest “expected fitness”. For juveniles, expected fitness is approximated as the expected probability of surviving and reaching reproductive size over a future time horizon. If none of the cells offer expected fitness higher than a size-dependent threshold, the fish migrates downstream out of its current reach. For adults, expected fitness is defined simply as survival until they spawn.

The following subsections explain the habitat selection and outmigration trait in detail.

4.2.1. Competition for resources via dominance hierarchy

The habitat selection trait assumes a size-based dominance hierarchy: fish can only use resources (food and velocity shelters) that have not been consumed by larger fish. Hughes (1992b) showed that stream salmonids rank feeding positions by desirability and the most dominant fish obtain the most desirable sites. Gowan and Fausch (2002) and Hughes (1992b) also showed that dominance is usually, but not always, proportional to length for. The hierarchy is implemented in *inSALMO* by executing the habitat selection method in order of descending fish length. The longest individual selects its cell first, and the food and velocity shelter it uses is subtracted from that available in the cell for additional salmon. Subsequent salmon therefore base their habitat selection not on the total resources in each cell but on the resources remaining unconsumed by larger fish.

4.2.2. Identify potential destination cells

When each individual salmon begins its habitat selection method, its first step is to identify the cells that are potential movement destinations. Distance and depth can limit potential destination cells; but the number of fish already in a cell does not limit its availability as a destination.

4.2.2.1. ***Distance limitation***

Only habitat cells within a certain distance are included as potential destinations. This maximum movement distance should be considered the distance over which a fish is likely to know its habitat well enough to be aware when desirable destinations are available, over a daily time step. The maximum movement distance should *not* be considered the maximum distance a fish could swim or migrate in a day.

The maximum movement distance is a function of length. Because mobility and spatial knowledge are assumed to increase rapidly with fish size, this distance is an exponential function. The parameters *fishMoveDistParamA* and *fishMoveDistParamB* are potentially site-specific: fish are likely to explore and be familiar with larger areas in lower-gradient rivers.

$$maxMoveDistance = fishMoveDistParamA \times fishLength^{fishMoveDistParamB}$$

In *inSALMO*, fish can follow a gradient toward better habitat if the gradient is detectable within the *maxMoveDistance*, but they do not have the ability to find and move toward some specific target if that target is beyond *maxMoveDistance*. For example, if habitat generally improves in an upstream direction, fish will have an incentive to gradually move upstream. However, if a very good location for some fish exists farther away than its *maxMoveDistance*, the fish will not be aware of it and try to move to it.

Movement observations from the literature cannot be considered direct measurements of *maxMoveDistance* but can be useful for evaluating its parameters. Observed movement distances in stream salmonids (e.g., Gowan and Fausch 1996, Harvey et al. 1999) show how far fish actually move, not the distance over which they evaluate habitat. These observations are also potentially confounded by a number of factors. Small fish may actually move more than large fish because they are less able to defend a location; this does not mean small fish have a larger maximum movement distance as defined in the model. Movement rates reported in the

literature are also potentially deceptive because they are rarely based on continuous or even daily observations of location.

However, literature observations do indicate that stream salmonids commonly select habitat over distances up to several hundred meters. Harvey et al. (1999) showed fall and winter movements of adult (18-24 cm length) cutthroat trout of up to about 55 m in one day in a moderate-gradient stream. Summer conditions (lower flows, higher metabolic rates and food requirements, higher population densities) may encourage greater movement distances. June (1981) observed little movement in newly emerged cutthroat trout <3 cm; dispersal started after they exceeded 3 cm in length.

Parameter values for a mid-sized, moderate-gradient stream (Table 4) estimate *maxMoveDistance* as 8 m for newly emerged juveniles with length of 4 cm, as 13 m for juveniles 5 cm long, and as 50 m for near-smolts 10 cm long. For adults, with lengths well over 50 cm, this formulation typically means that their entire reach is available for habitat selection.

Table 4. Example parameter values for fish movement distance.

Parameter	Definition	Value
fishMoveDistParamA	Multiplier for maximum movement distance (unitless)	50
fishMoveDistParamB	Exponent for maximum movement distance (unitless)	2

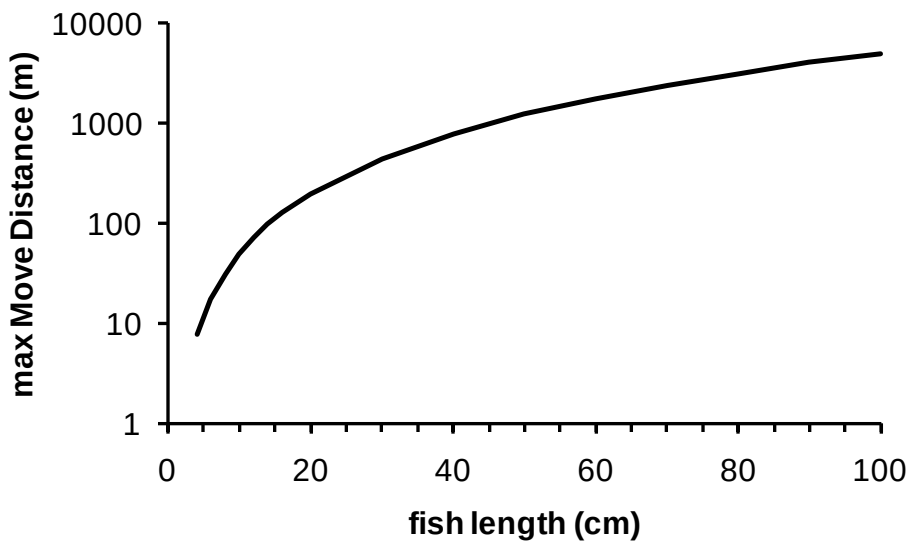


Figure 7. The maximum distance fish can move, as a function of their length, for *fishMoveDistParamA* = 50, *fishMoveDistParamB* = 2.

To identify potential destination cells for habitat selection, a model salmon first calculates its current *maxMoveDistance*. Then all cells in the salmon's reach with distance from the salmon's current cell (as defined in Section 2.2.6) less than *maxMoveDistance* are potential destinations.

4.2.2.2. ***Cells in other reaches***

Adult salmon cannot move to destinations outside the reach they were placed in when the model was initialize (for reasons explained in Section 4.1.2). Juveniles, though, can select habitat in other reaches. If, for example, *maxMoveDistance* for a fish is greater than the distance from the fish's current cell and the downstream end of its reach, and another reach is linked to the downstream end of the fish's reach, then some cells in the linked reach will be potential movement destinations.

The approach to identifying potential destination cells in adjacent reaches in *inSALMO* 1.0 is very simple: cells at the upstream and downstream ends of each reach are manually identified when habitat input is prepared, and using straight-line distances from cells to the nearest such end cell. Specifically:

- A new static cell variable is added to the input. This variable (*cellReachEnd*) has three possible values: "U" indicates that the cell is on the upstream end of the reach; "D" indicates that the cell is on the downstream end of the reach; and "I" means the cell is intermediate, not at either end. These values are assigned manually (typically, via the GIS) as part of preparing the cell input. Channel margin cells that are at one end of their reach but dry at normal flows can be given a *cellReachEnd* value of "I" to indicate that they should not be used to calculate distance to the end of the reach.

- Cells have two additional static variables that are calculated when the model is initialized. These variables (*cellDistToUS*, *cellDistToDS*) represent the distance from the cell to the upstream and downstream ends of its reach. These variables are simply set to the lowest straight-line distance from the cell's centroid to the centroid of any cell on the upstream and downstream ends of the reach.

- A fish determines whether its potential destination cells include some in other reaches by using *cellDistToUS* and *cellDistToDS*. For example, consider a fish in a reach that has a second reach below it, so that the downstream end of the fish's reach is connected to the upstream end of the other reach. The fish could potentially move into the downstream reach if its cell's value of *cellDistToDS* is less than the fish's value of *maxMoveDistance*. In that case, its potential destination cells would include those in the downstream reach with *cellDistToUS* less than (*maxMoveDistance* minus *cellDistToDS* of the fish's current cell). Potential destination cells are included from all reaches that are attached to the fish's current reach, at an end of that reach within *maxMoveDistance*. (However, a fish cannot move out of one reach, through a second, and into a third reach. Potential destination cells are obtained only from reaches adjacent to the fish's current reach.)

This approach is clearly not exact, especially for sharply curved reaches, but the high uncertainty in the distance over which fish sense and select habitat at a daily time scale makes the error in distance to reach ends unimportant.

For small fish, it is possible that no cells (other than its current one) are closer than *maxMoveDistance*. Having no potential destination cells poses an artificial barrier to movement, an artifact of the model's spatial resolution. This artifact could be important, for example by preventing newly emerged fish from moving from their natal redd to habitat where survival probabilities are higher. In such a situation, competition among newly emerged fish for food would largely be an artifact of the cell's size, which controls how much food is in it. To address this problem, a fish's potential destinations always include the cells adjacent to the fish's current cell. (These adjacent cells are identified as all cells sharing all or part of a side, or a corner, with the fish's current cell.) Cells from other reaches are not included among the adjacent cells that are always included as potential destinations.

4.2.2.3. ***Minimum depth***

Cells are excluded as destinations if they have depth ≤ 0. Fish are not required to move out of their current cell if its depth drops to zero, but the fitness measure they yse to evaluate potential destinations (Section 4.2.3) provides a strong incentive to move from dry cells. However, if the flow decreases so that the nearest cell with non-zero depth is farther away than a fish's maximum movement distance (not unlikely for very small fish), then the fish can be trapped in a dry cell. (See Section 4.4.3 concerning stranding mortality.)

4.2.3. Evaluate potential destination cells

Separate methods for selecting among the potential destination cells are used for juveniles, adults that have not spawned, and adults that have already spawned.

4.2.3.1. ***Juveniles***

Juveniles select the potential destination cell that provides the highest value of a fitness measure modified from the "expected maturity" measure of Railsback et al. (1999). The fitness measure represents the juvenile's expected success at surviving and growing until it can smolt and go to the ocean. This measure is calculated as:

$$expectedSmoltSuccess = nonstarvSurvival \times starvSurvival \times fracSmoltSize\,.$$

The variable *nonstarvSurvival* is the calculated probability of survival for all mortality sources except poor condition, over a specified time horizon given by the parameter *fishFitnessHorizon*. This method assumes that fish use a very simple prediction of future survival: that, over the time horizon, the daily survival probability for risks other than poor condition is equal to the current day's risks. The value of *nonstarvSurvival* is calculated as:

$$nonstarvSurvival = \left(S_i \times S_{ii} \times S_{iii} \ldots\right)^{fishFitnessHorizon}$$

where S_i, S_{ii}, S_{iii}, etc. are the daily survival probabilities for all the mortality sources (*i,ii,*…), evaluated for the current day, fish, and cell (these probabilities are described in Section 4.4). The value of *nonstarvSurvival* is determined for the fish's size *before* the daily growth that would occur at the potential destination cell; this assumption is made to simplify the model's software.

The formulation of *nonstarvSurvival* implicitly assumes that salmon consider all mortality sources in their habitat selection decision. This means that the salmon are assumed to be

aware of all the kinds of mortality in the model and are able to estimate the risk posed by each. This assumption seems reasonable for all the mortality sources currently in *inSALMO*.

In the equation for *expectedSmoltSuccess*, the value of *starvSurvival* is the probability of surviving the risk of poor condition (closely related to starvation; Section 4.4.4) over the number of days specified by the parameter *fishFitnessHorizon*. This term introduces the effects of food intake to the fitness measure. The value of *starvSurvival* is determined by the following steps (Railsback et al. 1999). The method assumes that fish evaluate *expectedSmoltSuccess* using the simple prediction that the current day's growth rate would persist over the time horizon.

- Determine the foraging strategy, food intake, and growth (g/d) for the fish and habitat cell in question, for the current day, using the methods in Section 4.3.

- Project the fish's weight, length, and condition factor *fishCondition* (Section 4.3.1) that would result if the current day's growth persisted over the fitness time horizon specified by *fishFitnessHorizon*. The daily growth is multiplied by *fishFitnessHorizon* to determine the change in weight over the time horizon; the corresponding change in length and *K* are determined using the methods described in Section 4.3.1.

- Approximate the probability of surviving starvation over the fitness horizon, estimated as as the first moment of the logistic function of poor condition survival vs. *K* (Section 4.4.4):

$$starvSurvival = \left[\frac{1}{a} \ln\left(\frac{1+e^{\left(aK_{t+T}+b\right)}}{1+e^{\left(aK_{t}+b\right)}} \right) \Big/ \left(K_{t+T} - K_{t}\right) \right]^{T}$$

  where K_t is the fish's value of *fishCondition* at the current day and K_{t+T} is the projected condition factor at the end of the fitness horizon, *T* is equal to *fishFitnessHorizon*, and *a* and *b* are the *logistA* and *logistB* variables (determined within the code from parameter values; see the logistic function conventions described in Section 2.2.7) for poor condition mortality. This equation would cause a divide-by-zero error when K_{t+T} equals K_t, a common condition because *K* equals 1.0 whenever fish are well-fed. This equation is also subject to significant errors due to the limits of computer precision when K_{t+T} is extremely close to K_t. To avoid these problems, *starvSurvival* is set equal to the daily survival probability for K_t, raised to the power *fishFitnessHorizon*, whenever the difference between K_{t+T} and K_t is less than 0.001.

The final term in the equation for *expectedSmoltSuccess* is *fracSmoltSize*, which represents how close to smolting a juvenile would be at the end of the fitness time horizon. It is simply (a) the length the fish is projected to be at the end of the time horizon, divided by (b) the parameter *fishOutmigrateSuccessL9*, and (c) limited to a maximum value of 1.0. The parameter *fishOutmigrateSuccessL9* represents a length at which outmigration success becomes high (Section 4.2.5).This term gives juvenile salmon an incentive to select cells with higher growth, encouraging them to reach the size necessary for smolting and ocean survival.

The time horizon variable *fishFitnessHorizon* is the number of days over which the terms of the expected smolt success fitness measure equation are evaluated. The biological meaning of this variable is the time horizon over which fish evaluate the tradeoffs between food intake and

mortality risks to maximize their probability of surviving and reproducing. It is discussed in the "unified foraging theory" (also called "dynamic state variable modeling" literature; Mangel and Clark 1986, Houston and McNamara 1999, Clark and Mangel 2000). Ideally, fitness is considered a lifetime process, so longer time horizons better reflect how an individual's fitness depends on how well it makes decisions throughout its reproductive life. However, the simple prediction used to evaluate *expectedSmoltSuccess*—that habitat and competitive conditions are constant over the time horizon—becomes very questionable for long time horizons. Smaller values of *fishFitnessHorizon* place less emphasis on food intake and avoiding starvation in movement decisions. Values of *fishFitnessHorizon* of 5 - 10 d cause *expectedSmoltSuccess* to vary almost exclusively with non-starvation survival, with very little effect of food intake and growth. Values of *fishFitnessHorizon* in the range of 100 d caused *expectedSmoltSuccess* to vary almost exclusively with growth rates when growth was less than the minimum needed to maintain a condition factor of 1.0. A value of 90 d has been successful in several studies validating related decision models (Railsback and Harvey 2002; Railsback et al. 2005).

4.2.3.1. ***Adults that have not yet spawned***

Adults that have not yet spawned use a habitat decision method similar to that of juveniles but with two key differences. First, adults do not consider as potential destinations any cells outside their current reach, for reason discussed at Section 4.1.2. Second, growth is assumed unimportant to adults so they select a cell that provides the highest value of a fitness measure that considers only survival:

$$expectedSurvival = nonstarvSurvival \times starvSurvival$$

where *nonstarvSurvival* and *starvSurvival* are as defined in Section 4.2.3.1. Hence, adults that have not yet spawned select habitat that provides a combination of safety from predators (e.g., deep cells with hiding cover nearby) and low energetic cost (low velocities or availability of velocity shelter, which reduce swimming metabolism and the risk of poor condition mortality).

4.2.3.2. ***Spawned adults***

After adults have actually spawned, they are assumed to remain in the cell they were in when they spawned. Female spawners actually move to the spawning cell during their spawning action, so they stay in the cell where their redd is (which is essential for making redd defense work; Section 4.1.2). Male spawners do not move to the cell they spawn in, so they remain in the cell they selected on the day they spawned.

4.2.4. Move to best destination and consume its resources

After each fish identifies the cell that has the highest value of its fitness measure, it moves there. When a fish moves into a cell, the resources it uses are subtracted from those available for subsequent fish (sections 3.2.3; 3.2.6). These resources may include one of the two kinds of food, and velocity shelter. Adults are assumed not to consume any food, and spawned adults are assumed not to use any velocity shelter. A fish may move into a cell even when none of these resources remain available to it, in which case its consumption of them is zero.

4.2.5. Outmigration as a habitat selection alternative

Juvenile salmon can "outmigrate"—move downstream toward the ocean—as an alternative to staying in their current reach. Outmigration is often thought of (and modeled, in previous

versions of *inSALMO*) as two different processes: voluntary and involuntary. Involuntary outmigration is thought of as the movement downstream, perhaps uncontrolled, of fish that have failed to establish and grow near where they were spawned. These fish may die, or they may end up in downstream habitat with sufficient resources for them to grow and survive. The widely observed downstream movement of moribund fry shortly after emergence is thought of as involuntary outmigration. Voluntary outmigration is considered the intentional movement toward the ocean of juvenile salmon that are relatively healthy and searching for “bigger” habitat as they grow or are nearing readiness to smolt and enter salt water.

In *inSALMO*, all outmigration is treated as the consequence of one part of the habitat selection decision: fish choose to move downstream if their expectation of success downstream is greater than their expected success in their current reach. Fish therefore decide to migrate out if they are either (a) doing poorly so expected success if they stay is low, or (b) approaching smolt size so their expected success if they outmigrate is high. This approach requires a definition of expected success at the current reach and expected success for downstream migration, and specification of how juveniles move downstream when they do.

Expected success in current reach. The measure of success in the fish’s current reach is simply its “expected smolt success” (the variable *expectedSmoltSuccess*; Section 4.2.3.1) at the best cell it considers as a habitat selection alternative. (This cell could be in a different reach than where the fish is when making its decision.)

Expected success downstream. The model of a fish’s expected reproductive success if it migrates downstream is a simple logistic function of the fish’s length. This “outmigration success” function (Figure 8) is not a model of how likely the fish is to survive to adulthood and reproduce but instead a model of the *relative* likelihood of eventual reproductive success compared to the alternative of staying in the best cell available to it. The logistic form and parameters of this function are based on the assumption that the probability of a smolt returning to spawn increases sharply with the length at which it smolts, over a range of approximately 5-12 cm, but is little affected by smolt size above that range. A second basis for the logistic function is the assumption that increasing size increases the probability that a downstream migrant can find and exploit profitable habitat while reducing the probability of predation by other fish.

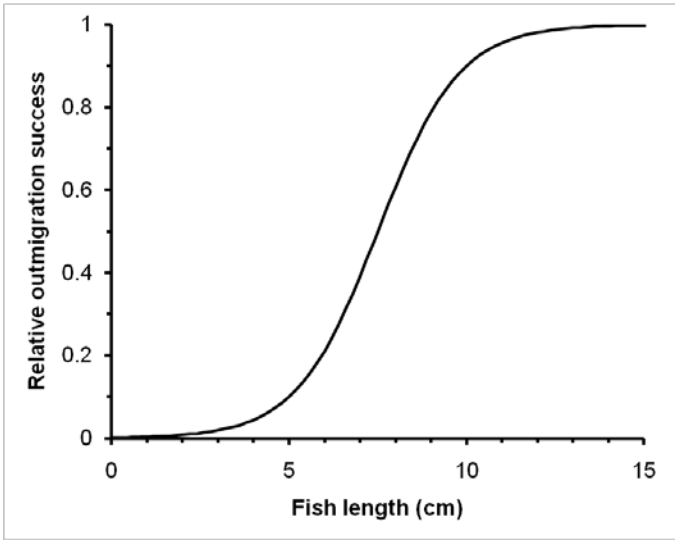


Figure 8. The outmigration success function of fish length, with *fishOutmigrateSuccessL1* and *fishOutmigrateSuccessL9* set to 5.0 and 10.0 cm.

The outmigration success function is defined by two new fish parameters: *fishOutmigrateSuccessL1* and *fishOutmigrateSuccessL9*. These are, respectively, the lengths (cm) at which the function has values of 0.1 and 0.9.

Outmigration movement. When a fish chooses to migrate downstream, it moves, on the current day, at least to the next-downstream reach if there is one. Upon arriving at that reach, the fish repeats its habitat selection action to either find a good cell in that reach or move downstream again. If instead there is no other model reach downstream, the fish is treated as an outmigrant from the model.

To provide complete detail, the following steps implement outmigration as part of habitat selection.

- The fish executes habitat selection as described in Section 4.2.3.1, calculating *expectedSmoltSuccess* for each potential destination cell and identifying the cell with highest *expectedSmoltSuccess*.

- The fish then evaluates the logistic outmigration success function for its current length.

- If *expectedSmoltSuccess* for the best cell is greater or equal to the outmigration success function value, the fish moves to that cell (Section 4.2.4).

- If instead *expectedSmoltSuccess* for the best cell is less than the outmigration success function, the fish migrates downstream by:

    - Identifying any other reaches that are immediately downstream (its reach's list of other reaches which have their upper end connected to its downstream end);

- If there are no such downstream reaches, “migrating out” of the model as described in Section 4.2.6.
- Creating a list of all cells that are at the upstream end of the downstream reaches (as described in Section 4.2.2.2) and currently have depth greater than zero;
- Moving to a cell selected randomly from the above list, removing itself from its current cell;
- Repeating the entire habitat selection action, which could result in the fish finding a profitable cell in its new reach or in moving on downstream to the next reach or out of the model.
- If there are one or more downstream reaches but none of their upstream-end cells have depth greater than zero, then the fish remains in the best available cell of its current reach instead of migrating downstream. (In this unexpected event, the code issues a warning statement.)

An evaluation of this approach to outmigration indicates that it produces expected characteristics (Figure 9). When juvenile salmon are small (6 cm and less) they migrate downstream only if growth in their current reach is negative (or if other risks are extremely high). At intermediate lengths (e.g., 8 cm) fish migrate out unless their reach offers low risk and zero to high growth. At lengths approaching 10 cm fish migrate downstream unless their current habitat is extremely safe and offers positive growth. By the length of 12 cm (not shown in Figure 9) all fish have migrated downstream.

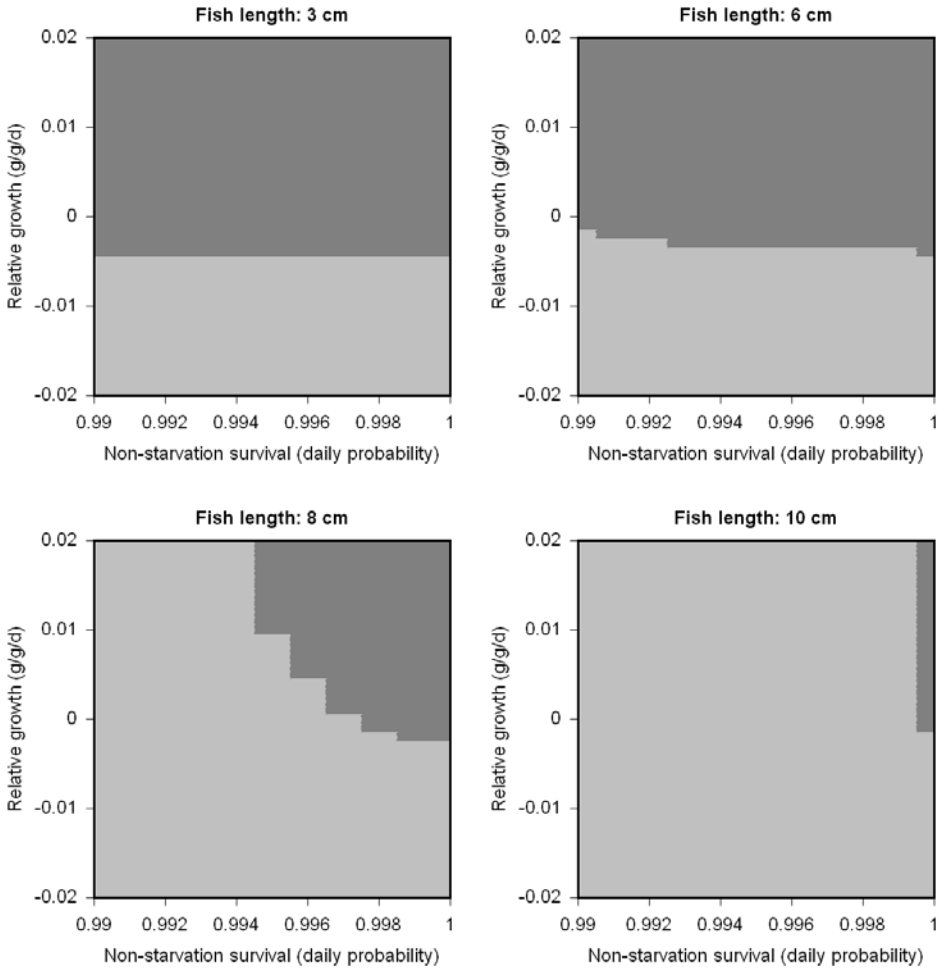


Figure 9. Results of the outmigration decision method with *fishOutmigrateSuccessL1* equal to 5.0 and *fishOutmigrateSuccessL9* at 10.0 cm. Light grey regions indicate combinations of growth and risk conditions under which juvenile salmon migrate downstream, and dark grey regions indicate where salmon remain in the current reach. The X axis is the daily probability of surviving factors other than starvation (e.g., predation, high temperature). The Y axis is daily growth rate as grams of growth per gram of fish weight. The four panels show how results depend on fish length. Fish were assumed to currently be in good condition; lower condition (weight at length) slightly increases the tendency to migrate downstream.

4.2.6. Removal of outmigrants

Juvenile fish that have migrated downstream beyond the downstream-most reach are treated as “outmigrants”. They are removed from the list of live fish and placed on a separate list of outmigrants, and removed from the last habitat cell they occupied. No actions are executed on outmigrants, so they do not change for the rest of the model run. On the day they decide to migrate, outmigrants are not subject to mortality and hence cannot die and outmigrate on the same day.

4.3. Feeding and Growth

4.3.1. Overview

This section describes the methods for determining the daily growth—change in weight and length—that a fish obtain in their habitat cell. These methods are used both in the habitat selection decision to determine how much growth a fish would obtain in each cell it considers as a potential destination, and to simulate growth (the third daily action by fish; Section 8.2). This first subsection provides an overview of the feeding and growth methods, listing the major assumptions. Full detail is provided starting with Section 4.3.2.

Adult salmon follow all the methods for feeding and growth described in this section, but with one major difference: their food intake is always assumed to be zero. Because they still have metabolic costs, adults therefore gradually lose weight. This assumption, combined with the decision of how to feed (Section 4.3.9) and the habitat selection methods (Section 4.2) cause unspawned adults to select feeding behaviors and habitat cells that provide low rates of weight loss and high survival probabilities.

The feeding and growth formulation of *inSALMO* is conceptually related to a number of other models. First, it borrows both basic concepts and detailed methods from the extensive fish bioenergetics literature. The concepts of (1) modeling growth as net energy intake, the difference between energy input from food and energy consumption for metabolism; and (2) modeling metabolic energy consumption as a function of fish size, swimming speed, and temperature; are well-established and tested (to some extent) in the literature (Hanson et al. 1997; see also Brandt and Hartman 1993, Elliott and Hurley 2000). Second, bioenergetics models and feeding models have been combined to predict net energy intake as a function of fish size and habitat conditions (especially, depth and velocity) by a number of researchers (e.g., Fausch 1984, Hughes and Dill 1990, Hill and Grossman 1993, Braaten et al. 1997, Van Winkle et al. 1998, Hayes et al. 2000, Gowan and Fausch 2002, Grossman et al. 2002).

One important characteristic of *inSALMO* is that competition among individual fish for food is modeled. A fish's food intake is assumed to be limited by either the availability of food or the ability of the fish to capture food. The ability to capture food depends on fish size (increasing with length, because larger fish see and swim better) and on habitat conditions such as velocity and depth in the fish's cell. Food availability depends on how much food is produced in the cell and how much is consumed by competing fish (Section 3.2.6).

Fish in *inSALMO* are assumed to always feed during daylight hours and never at night, a major simplifying assumption. While salmonids have long been thought of as feeding visually and therefore during day, recent literature shows that night feeding is not unusual and under some conditions is more common than daytime feeding (e.g., Fraser and Metcalfe 1997, Metcalfe et al. 1999, Bradford and Higgens 2001). Whether an individual feeds during day or night (or neither) appears to emerge from how mortality risk and food intake vary between day and night, which can in turn vary with fish size, competition, and many habitat variables. How trout choose between feeding during day and night has been simulated successfully in a trout model similar to *inSALMO*, (Railsback et al. 2005), but this capability requires a major increase in the model's complexity. This additional complexity does not appear justified by the objectives of *inSALMO*. While the assumption that salmon feed during daytime only is clearly not always realistic, it is useful for the purposes that *inSALMO* is intended for.

inSALMO does not specify the exact kinds of food consumed by fish, but its feeding formulation and parameters generally represent invertebrate food, not other fish.

Fish in *inSALMO* can use either of two feeding strategies. Drift feeding, in which the fish remains stationary and captures food as it is carried past by the current, is the most studied and often the most profitable strategy (Fausch 1984, Hill and Grossman 1993, Hughes and Dill 1990). Drift food intake is modeled as a function of stream depth and velocity and fish length; intake peaks at an optimal velocity that is higher for larger fish. Drift intake decreases as turbidity increases, as turbidity makes it harder for fish to detect food items. Metabolic costs for drift feeding increase with water velocity, but use of velocity shelters reduces this cost. The second feeding strategy is active searching for food. Search feeding can be important when competition for food is intense, conditions for drift feeding are poor, or the abundance of benthic food is high (Nielsen 1992, Nislow et al. 1998). The energetic benefits of search feeding are assumed to be mainly a function of food availability, with energetic cost depending on water velocity.

The feeding and growth methods calculate the potential food intake and metabolic costs a fish would experience in a cell, for both drift and search feeding. Standard bioenergetics approaches (Hanson et al. 1997) are used by *inSALMO* to calculate net energy intake (the difference between energy intake from food and metabolic energy costs; net energy is often negative) for each feeding strategy. The fish then selects the strategy that provides the highest net energy intake. Growth (increase in body weight, g/d) is proportional to net energy intake.

From a fish's daily growth, its length and condition factor are updated. How an organism allocates its energy intake to growth (increase in length), storage (increase in weight or fat reserves but not length), or gonads is in reality a complex, adaptive decision. For example, a juvenile fish may reduce its risk of predation most by increasing in length as rapidly as possible, but allocating all energy intake to growth instead of storage increases the risk of starvation during periods of reduced intake. However, *inSALMO* does not model energy allocation as an adaptive trait. Instead it uses the approach of Van Winkle et al. (1996) that simply forces fish to maintain a standard relation between length and weight during periods of positive growth.

The method for calculating daily change in length adopted from Van Winkle et al. (1996) also uses their nonstandard definition of a condition factor. In fisheries science, a condition factor is a unitless index of a fish's weight relative to its length. A higher condition factor indicates that a fish is heavy for its length and has high energy reserves, and therefore less vulnerable to starvation or disease during periods of negative growth. The condition factor variable used in *inSALMO* (*fishCondition*) can be considered the fraction of "healthy" weight a fish is, given its length. The value of *fishCondition* is 1.0 when a fish has a "healthy" weight for its length, according to a length-weight relation input to the model via fish parameters *fishWeightParamA* and *fishWeightParamB*:

$$fishHealthyWeight = fishWeightParamA \times fishLength^{fishWeightParamB}.$$

Fish grow in length whenever they gain weight while their value of *fishCondition* is 1.0. Condition factors less than 1.0 indicate that the fish has lost weight. In this formulation, values of *fishCondition* cannot be greater than 1.0. Weight (*fishWeight*, g), length (*fishLength*, cm), and *fishCondition* are calculated in this way.

- The fish's new weight is determined by adding its daily growth (which can be negative) to its previous weight.

- The fish's new weight is used, with the inverted length-weight relation for healthy fish, to calculate *fishWannabeLength*, the length the fish would be if its condition factor were 1.0:

$$fishWannabeLength = \left(\frac{fishWeight}{fishWeightParamA} \right)^{1/fishWeightParamB} .$$

- If the fish's current length is less than *fishWannabeLength* (indicating that the fish is not underweight), then its new length is set to *fishWannabeLength*. The fish grows in length while keeping its *fishCondition* value equal to 1.0.

- If the fish's current length is greater than *fishWannabeLength* (indicating that the fish is underweight for its length), its length is not changed.

- The new value of *fishCondition* is equal to the fish's new weight divided by the "healthy" weight for a fish its length:

$$fishCondition = fishWeight \Big/ \left(fishWeightParamA \times fishLength^{fishWeightParamB} \right).$$

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

- Fish cannot store a high-energy-reserve condition. Fish will have a condition of 1.0 *only* on those days when daily growth is positive. Even if a fish has eaten well for many days in succession, its *fishCondition* can only be as high as 1.0 and one day of negative net energy intake causes condition to fall below 1.0. This could be important under conditions of highly variable food intake because survival is assumed to decrease with condition (Section 4.4.4).

- This weight-based condition factor is not the best predictor of starvation mortality (Section 4.4.4).

- This formulation locks in a length-weight relationship for growing fish. Calibration of growth to situations where this relationship is valid will be automatic, but calibration to situations where the relationship is not valid will be impossible. For example, *inSALMO* cannot predict the existence of unusually fat fish.

These limitations could be eliminated only by making *inSALMO* considerably more complex. Methods for representing energy allocation more realistically in IBMs have not yet been developed and tested. The current formulation appears adequate and appropriate for *inSALMO*'s objectives.

Example parameter values for the length-weight relationship are provided in Table 5. These parameters should not simply be regression parameters calculated from observed data; they must describe a site-specific length-weight relation for fish in good condition.

Table 5. Example parameter values for the length-weight relation, for length in cm and weight in g.

Species and site	Parameter	Value
Juvenile Chinook salmon, Sacramento River (Petrusso and Hayes 2001)	*fishWeightParamA*	0.00411
	fishWeightParamB	3.49

4.3.2. Activity budget

Energy intake and costs differ between feeding vs. resting fish. Energetic calculations are based on hourly energy rates (j/h), and the daily energy totals depend on how many hours are spent feeding vs. resting.

In *inSALMO*, salmon are assumed to spend all daylight hours feeding and all night hours resting. Daylight hours are assumed to include one hour before sunrise and one hour after sunset. Consequently, the time spent feeding per day (*feedTime*, h/d) is *daylength* + 2.

4.3.3. Food intake: drift feeding strategy

Drift feeding fish wait and capture invertebrates as they are carried within range by the current. The drift feeding energy intake formulation of *inSALMO* is unique but conceptually related to the previous feeding and net energy intake models cited in Section 4.3.1. This literature shows clearly that the distance over which fish can see and capture food increases with salmon size and decreases with water velocity. Unlike previous models, *inSALMO* includes the negative effect of turbidity on the ability of salmon to see and capture prey. Turbidity can vary dramatically among sites and over time, and its effects on salmon feeding are strong and relatively predictable. Unlike some previous models of drift feeding, *inSALMO* neglects prey size as a variable. Prey size is naturally variable and unpredictable, and its effects could not be easily be distinguished from those of other factors.

Drift-feeding fish are assumed to capture some of the food items that pass within a “capture area” (*captureArea*, cm^2), a rectangular area perpendicular to the current, the dimensions of which depend only on fish size (explained below). The fraction of food items passing through the capture area that are actually caught (*captureSuccess*, unitless) decreases with cell velocity, increases with fish swimming ability, and decreases with turbidity. A fish’s intake rate (*driftIntake*, g/h) is calculated as the mass of prey passing through the capture area times the capture success:

$$driftIntake = captureSuccess \times habDriftConc \times velocity \times captureArea \times 3600.$$

In this equation, *habDriftConc* (g/cm^3) is a habitat reach variable (Section 3.1.1) and the last term (3600 s/h) converts the rate from per second to per hour.

A detection distance approach is used to calculate *captureArea*. Detection distance is defined as the distance over which fish can see and attack—but not necessarily capture—prey. Detection distance is believed to depend primarily on the size of the fish (bigger fish have bigger, more sensitive eyes) and the size of the prey (bigger prey being easier to detect). Schmidt and O'Brien (1982) collected empirical data on how detection distance in a stream salmonid (arctic grayling) varied with fish and prey size. These experiments used zooplankton as prey, but their results have been used successfully as the basis of drift feeding models of Hughes (1992a) 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 10). 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 than a line does. The linear model shown in Figure 10 was chosen for several reasons. First, it captures the fact that very small salmon cannot use as wide a range of prey sizes as larger fish can, a process not otherwise represented in the feeding model. Second, a logarithmic fit to these data predicts negative detection distances for salmon 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 *fishDetectDistanceParamA* and *fishDetectDistanceParamB*, defined below). The pre-calibration analysis indicated that the growth rates of very small salmon are very sensitive to the intercept. An intercept of 4.0 was found to provide growth of very small salmon that was realistic at the same drift food availability values that produce realistic growth rates in larger fish.

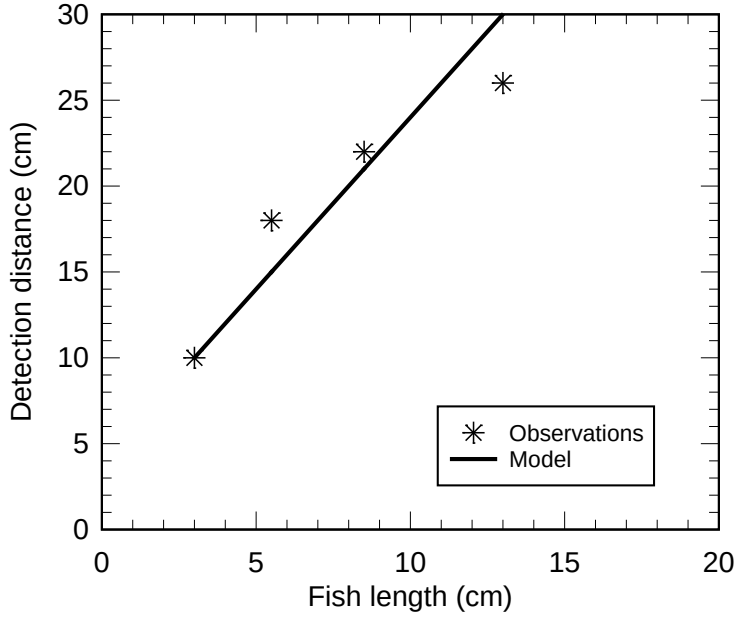


Figure 10. Relation between fish length and prey detection distance observed by Schmidt and O'Brien (1982), for arctic grayling feeding on 0.2 cm zooplankton.

Detection distance is adjusted for turbidity. 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 decrease. Barrett et al. (1992) attempted to evaluate the effect of turbidity on the ability of trout to detect and capture drift food, but their experiment had several weaknesses. The experiment used shallow depths and prey that floated on the surface, likely increasing the fish's ability to detect prey. More importantly, fish and prey were confined to a relatively narrow channel, limiting the ability to measure effects of low turbidity levels because detection distance could exceed the channel width. Sweka and Hartman (2001) conducted a similar experiment but with fewer limitations due to the experimental apparatus. This experiment included a fairly clear test of the effects of turbidity on the ability of fish to detect prey, over a range of 3-40 NTUs. Sweka and Hartman (2001) 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 *inSALMO* for <u>relative</u> detection distance (the fractional reduction in detection distance due to turbidity, at turbidity levels above zero) is based on the data of Sweka and Hartman (2001) but differs from their curve in two ways.

First, *inSALMO* assumes that turbidity has no effect at values below a threshold of 5 NTUs (defined by the parameter *fishTurbidThreshold*). 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 clearly shows an effect of turbidity at levels below 5 NTUs (see, e.g., Figure 11), and it seems likely that below such a threshold reactive distance is limited by other factors such as turbulence and the ability (or net benefit) of catching food items very far away. Another reason for assuming a turbidity threshold is to avoid

making *inSALMO* highly sensitive to low turbidity levels, which are hard to measure or estimate accurately.

The second change is adding 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. This conclusion is also supported by unpublished studies at Humboldt State University (S. Hadden, unpublished data) which show trout confined to narrow channels able to capture some drift at turbidity levels exceeding 70 NTUs. Therefore, *inSALMO* includes a parameter *fishTurbidMin* which limits the effect of turbidity on detection distance (Figure 11).

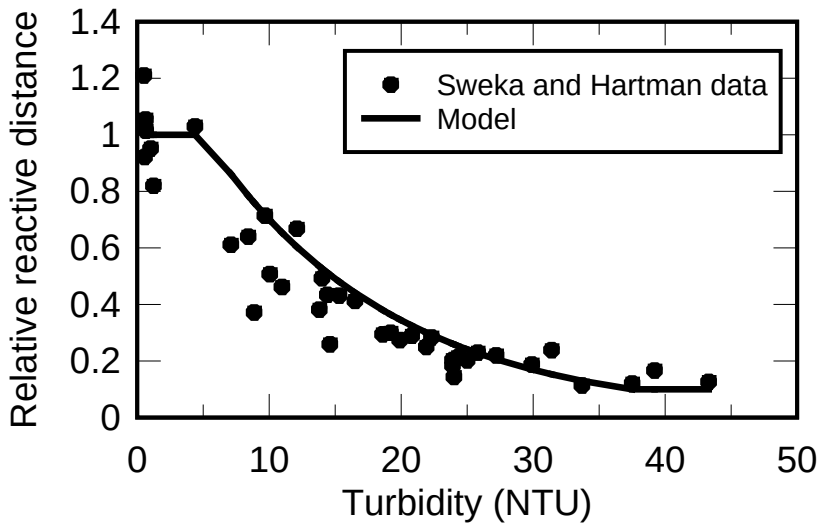


Figure 11. Relative detection distance vs. turbidity: model and data of Sweka and Hartman (2001) used to fit the model.

Detection distance is therefore modeled with this equation:

$$detectDistance = \left[fishDetectDistParamA + \left(fishDetectDistParamB \times fishLength \right) \right] \times turbidityFunction$$

where:

turbidityFunction = 1.0 if *habTurbidity* <= *fishTurbidThreshold*

else *turbidityFunction* =

$$\max\left[\exp\left(fishTurbidExp \times \left(habTurbidity - fishTurbidThreshold \right)\right), fishTurbidMin\right].$$

Parameter values are in Table 6 and the resulting model in Figure 11. The value of *fishTurbidExp* was fit via regression to the data of Sweka and Hartman (2001), who measured absolute reactive distance vs. turbidity by: (1) Establishing the reactive distance for negligible turbidity as the mean of reactive distances observed at turbidities less than 5 NTU; the seven

such observations had a mean reactive distance of 80.8 cm. (2) Calculating the relative reactive distance for other observations as the observed reactive distance divided by 80.8. (3) Using exponential regression on relative reactive distance vs. (turbidity - 5 NTU); the regression line was forced through the point (0,1) so relative reactive distance is one when turbidity is 5.

Several previous salmonid feeding models assumed that the capture area is a circle or half-circle with radius equal to the detection distance, but Booker et al. (2004) show that failing to consider depth (which often is less than the detection distance) can cause major errors. *inSALMO* uses a capture area for drift feeding that depends on the detection distance and cell depth. The width of the rectangular capture area is twice the detection distance: fish are assumed able to detect all drift that comes within the detection distance to their left and right, as they face into the current. The height of the capture area is the minimum of the reactive distance and the depth, as fish are assumed more likely to be near the stream bottom than at mid-depth when feeding:

$$captureArea = [2 \times detectDistance] \times [\min(detectDistance,\ cellDepth)].$$

While the capture area represents the area over which drift-feeding salmon can detect prey, capture success represents what fraction of detected prey are actually caught. Capture success is largely a function of water velocity. Fish must be able to swim to the prey, capture it, and return to their feeding station. At higher velocities, maneuvering quickly enough to capture prey is more difficult, and swimming longer distances after prey requires more energy (because the fish must swim back upstream to return to their feeding station; Hughes et al. 2003). Capture success is also affected by temperature, as the ability of fish to maneuver and swim rapidly is reduced at low temperatures.

Hill and Grossman (1993) 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 velocities ranging from 0 to 40 cm/s. Capture success was evaluated as the fraction of prey caught, within the fish's detection distance. Hill and Grossman (1993) approximated the detection distance as 2.5 times the fish's standard length, which is fairly close to the detection distance used in *inSALMO* (Figure 10). Hill and Grossman measured capture success within each of three ranges: the inner 20% of the capture distance, 20-60% of capture distance, and 60-100% of capture distance. To develop parameters for *inSALMO*, these values were averaged over the entire capture distance. For all the observations (35 combinations of fish size, temperature, and water velocity), capture success fit a logistic function of the ratio of water velocity to maximum sustainable swimming speed of the fish (Figure 12). (Maximum sustainable swimming speed is a function of fish length and water temperature. The method for modeling it is presented in Section 4.4.2.) Maximum sustainable swim speed (*fishMaxSwimSpeed*) appears to be useful for modeling capture success for two reasons: first, it scales capture success with both fish length and temperature. Second, Hughes et al. (2003) observed that large brown trout actually swim at sustainable (or even lower) speeds when capturing food.

$$captureSuccess = \mathrm{logistic}\left(habVelocity / fishMaxSwimSpeed\right).$$

Parameters for this logistic function are in Table 6.

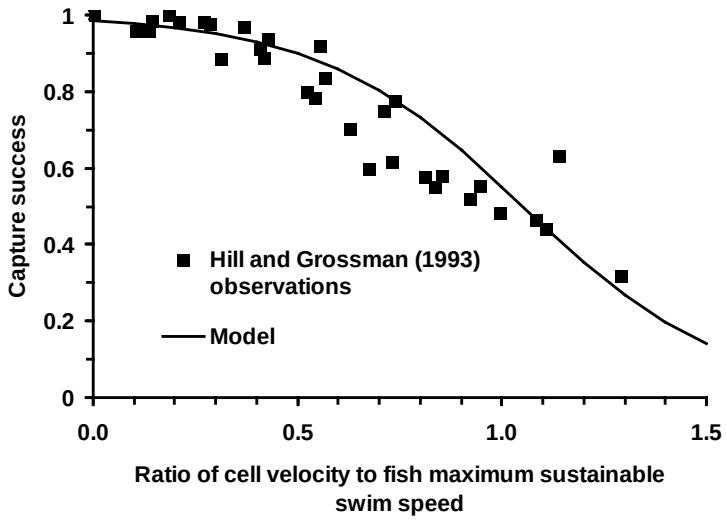


Figure 12. Capture success model and the laboratory observations it was based on.

Table 6. Detection distance and capture success parameters.

Parameter	Definition	Recommended value
fishDetectDistParamA	Intercept in equation for detection distance (cm)	4.0
fishDetectDistParamB	Multiplier in equation for detection distance (unitless)	2.0
fishTurbidThreshold	Highest turbidity that causes no reduction in detection distance (NTU)	5.0
fishTurbidExp	Multiplier in exponential term for the turbidity function (unitless)	-0.0711
fishTurbidMin	Minimum value of the turbidity function (unitless)	0.1
fishCaptureParam1	Ratio of cell velocity to fish's maximum swim speed at which capture success is 0.1 (unitless)	1.6
fishCaptureParam9	Ratio of cell velocity to fish's maximum swim speed at which capture success is 0.9 (unitless)	0.5

4.3.4. Food intake: active searching strategy

Actively searching for benthic or drop-in food is an alternative to the drift-feeding strategy. Unlike drift feeding, there are no established models for search feeding by salmon. An optimal foraging approach would be to assume fish search for food at a rate that maximizes the difference between energy intake from feeding and energy cost of swimming. To avoid the complexity of such an approach, *inSALMO* simply assumes that the rate of search food intake is proportional to the rate at which search food becomes available: every fish 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 fish's maximum sustainable swim speed. This velocity function 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 formulation; Section 4.3.7.)

The search food intake model is:

$$searchIntake = habSearchProd \times fishSearchArea \times \max\left(\left[\frac{fishMaxSwimSpeed - cellVelocity}{fishMaxSwimSpeed}\right], 0\right)$$

where *searchIntake* (g/h) is the rate at which food is taken in via search feeding, *habSearchProd* (g wet weight/h-cm^2) is the rate at which search food is produced (Section 3.2.6), *fishMaxSwimSpeed* is the fish's maximum sustainable swimming speed (cm/s; Section 4.4.2), and *cellVelocity* (cm/s) is the velocity of the fish's cell. The proportionality constant *fishSearchArea* (cm^2) can be loosely interpreted as the area over which the production of stationary (non-drifting) food is consumed by one fish. This search area is not necessarily a contiguous piece of stream area: a small fish searching a small area closely may obtain the same food intake as a big fish spot-searching over a much larger area. Because *habSearchProd* and *fishSearchArea* have the same effect on search intake and both would be very difficult to measure, either would be a good parameter to use for calibration. Note that fish size does not affect search food intake except for the effect of size on *fishMaxSwimSpeed*; therefore, search feeding is more likely to be the desirable strategy for smaller fish.

Note that turbidity is not assumed to affect search feeding. While search feeding can sometimes be primarily visual, anecdotal evidence (e.g., observations of trout with full stomachs, foraging along stream margins during extremely turbid flood flows) indicate that salmonids can search-feed successfully using other senses. DeRobertis et al. (2003) conducted tank experiments resembling search feeding by juvenile chum salmon, observing feeding success at various turbidity levels. Feeding success under daytime conditions did not decrease consistently at turbidities between zero and 20 NTU; at 40 NTU feeding success was about one third of that in clear water. (During nighttime light levels, even turbidities up to 40 NTU caused no decrease in feeding success.) Because the effects of turbidity on search feeding are apparently limited, they are ignored in *inSALMO*.

4.3.5. Food intake: maximum consumption

As part of the net energy intake calculations, calculated food intake from drift or search feeding is checked to make sure it does not exceed the physiological maximum daily intake. This maximum daily consumption, referred to as *cMax* (g/d) in the bioenergetics literature, represents the maximum rate of food consumption if a fish is limited only by its physiology. Field bioenergetics studies (Preall and Ringler 1989, Railsback and Rose 1999) indicate that actual food intake does not approach *cMax* under typical conditions. However, here *cMax* serves the purpose of restricting intake and growth during low temperatures, a function otherwise lacking in the model (except that the time spent feeding becomes zero at temperatures below a threshold; Section 4.3.2). Cunjak et al. (1998) cite evidence that low food assimilation efficiencies and gut evacuation rates, which can be represented by *cMax*, limit energy intake in cold temperatures.

Unfortunately, *cMax* 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). However, there are a number of published equations for *cMax* that include (a) an allometric function, relating *cMax* to fish size; and (b) a temperature function (Hanson et al. 1997). The equation used in *inSALMO* is:

$$cMax = fishCmaxParamA \times fishWeight^{(1+fishCmaxParamB)} \times cmaxTempFunction.$$

This equation is widely used with the parameters developed by Rand et al. (1993) for rainbow trout (Table 7) for modeling *cMax* of salmonids in general (e.g., Van Winkle et al. 1996, Railsback and Rose 1999, Booker et al. 2004).

The *cMax* temperature function used in *inSALMO* 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 *cMax* at 10, 14, 19, 22, and 25ºC. Previous models of *cMax* 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 5-22ºC; and of Elliott (1982) who studied brown trout. Instead of an equation, the *cMax* temperature function is a set of seven points used to interpolate a value of *cmaxTempFunction* from the temperature of a fish's habitat reach (Table 8).

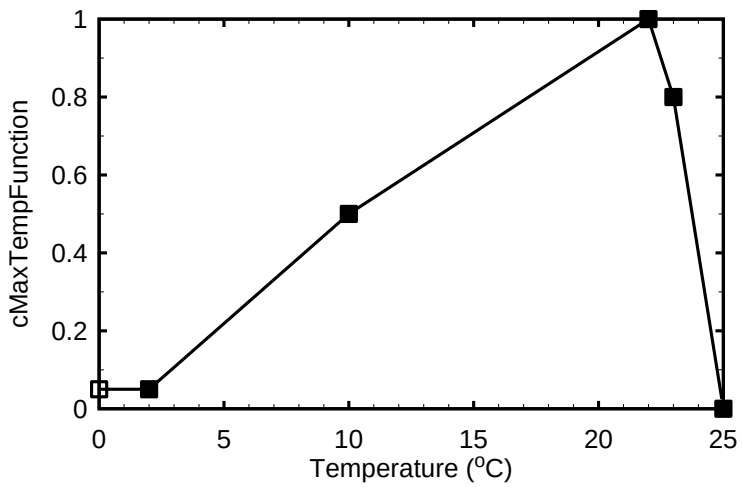


Figure 13. Temperature function for *cMax*.

While several sets of equations and parameters for *cMax* have been published for different salmonid species, careful scrutiny of these publications indicate that the differences in models of *cMax* are more likely to result from differences in experimental methods than from differences among species or stocks. Considering the inherent uncertainty in *cMax* and its limited effect on results of *inSALMO*, the parameters in Table 7 and Table 8 are cautiously recommended for all stream salmonid species.

Table 7. Parameter values for allometric function of maximum consumption.

Parameter	Definition	Value
fishCmaxParamA	Allometric constant in *cMax* equation (unitless)	0.628
fishCmaxParamB	Allometric exponent in *cMax* equation (unitless)	-0.3

Table 8. Parameter values for temperature function of maximum consumption. Each row in the table defines one of the points in Figure 13.

Parameter Name	Temperature (°C)	Parameter Name	Temperature Function Value (unitless)
fishCmaxTempT1	0	*fishCmaxTempF1*	0.05
fishCmaxTempT2	2	*fishCmaxTempF2*	0.05
fishCmaxTempT3	10	*fishCmaxTempF3*	0.5
fishCmaxTempT4	22	*fishCmaxTempF4*	1.0
fishCmaxTempT5	23	*fishCmaxTempF5*	0.8
fishCmaxTempT6	25	*fishCmaxTempF6*	0
fishCmaxTempT7	100	*fishCmaxTempF7*	0

4.3.6. Food intake: daily food availability

The food intake of each fish can be limited by the total amount of drift (*driftDailyCellTotal*, g/d) and search (*searchDailyCellTotal*, g/d) food available each day in its cell. These daily food availability values are a function of the fish's feeding time (*feedTime*, h; Section 4.3.2) because food produced during non-feeding hours cannot be considered available to the fish. The daily food availability rates are calculated from the hourly food availability rates described in Section 3.2.6.2. The hourly availability rates are the rate at food is produced in the cell, minus food consumption by larger fish. Therefore, hierarchical competition for food is implemented via the food availability rates. Daily food availability for a fish is determined as:

$$driftDailyCellAvail = driftHourlyCellAvail \times feedTime$$

and:

$$searchDailyCellAvail = searchHourlyCellAvail \times feedTime$$

where *driftHourlyCellAvail* and *searchHourlyCellAvail* are as defined in Section 4.3.2.

4.3.7. Respiration costs and use of velocity shelters

Conventional bioenergetics modeling approaches for fish (Hanson et al. 1997) model respiration as the energetic cost of metabolism and swimming. This approach is adopted for *inSALMO*, modeling (a) standard respiration that is independent of the fish's activity, and (b) an additional activity respiration that increases with the daily swimming speed.

Swim speeds. Drift-feeding fish are assumed to swim at a speed (*swimSpeed*, cm/s) equal to their habitat cell's water velocity unless they have access to velocity shelter. Fish using the search feeding strategy are assumed to swim at a speed equal to their cell's mean water

velocity. These two assumptions are a highly simplified representation of how real salmon swim within a day, but the consequent error in respiration costs is neglected instead of making the model more complex.

If a drift-feeding fish has access to velocity shelter, then its *swimSpeed* is assumed equal to a constant fraction of its habitat cell's mean water velocity. This fraction is defined by the habitat parameter *habShelterSpeedFrac*. 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., Baltz and Moyle 1984, Baltz et al. 1987, Moyle and Baltz 1985). However, relations between focal and depth-averaged velocities observed in these studies are not directly applicable to *inSALMO* because *habShelterSpeedFrac* approximates the difference between cell average water velocity and the swimming speed of a fish using velocity shelter. The best value of this parameter will vary with the kind of velocity shelter being used and could easily be estimated in the field by using a velocity meter. 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 *habShelterSpeedFrac*. For the Green River, Utah, where substrates are relatively small and embedded, a value of 0.5 was used (Railsback et al. 2005).

Velocity shelter access. Model salmon are assumed to compete for available velocity shelter space, similar to how they compete for available food. The following steps determine whether each fish has access to shelter in a habitat cell.

- Each cell has a limited area of velocity shelter; this area varies among cells but is constant over time (Section 3.2.3).
- Each drift-feeding fish is assumed to use up an area of velocity shelter equal to the square of its length.
- A fish has access to velocity shelter in a cell only if the sum of shelter areas occupied by larger drift-feeding fish in the cell is less than the cell's total shelter area.

Each fish is assumed to use only a small shelter area (the square of its length) to ensure that fish compete with each other for food, not for shelter area, unless velocity shelter clearly limits net energy intake.

Respiration cost model. *inSALMO* uses the Wisconsin Model equation 1 for respiration (Hanson et al. 1997), as modified by Van Winkle et al. (1996) to apply the activity respiration rate only during active feeding hours. The parameters that Rand et al. (1993) developed for steelhead trout (converted from calories to joules; Table 9) are widely used and appear to be the best available for stream salmonids in general. This formulation breaks respiration into two parts: standard respiration (*respStandard*, j/d) takes place 24 h/d and includes no effect of activity; activity respiration (*respActivity*, j/d) is the energy needed to swim during feeding. Total respiration (*respTotal*, j/d) is the sum of these two. The equations are:

$$respTotal = respStandard + respActivity \,,$$

$$respStandard = \left(fishRespParamA \times fishWeight^{fishRespParamB} \right) \times \exp\left(fishRespParamC \times temperature \right)$$

and

$$respActivity = \left(\frac{feedTime}{24} \right) \times \left[\exp\left(fishRespParamD \times swimSpeed \right) - 1 \right] \times respStandard \ .$$

Data collected by Myrick (1998; see also Myrick and Cech 2000) indicate that the standard respiration formulation overestimates the effect of temperature on respiration rates and does not account for a decrease in respiration observed at temperatures above 22°. Because of the Wisconsin Model equation's exponential temperature function, these problems cannot be fixed by changing parameter values. However, realistic calibrations of growth have been made with this formulation. The decrease in respiration by inactive fish at high temperatures observed by Myrick (1998) in laboratory respiration chambers may not be applicable in natural settings.

Table 9. Parameter values for respiration.

Parameter	Definition	Units	Value
fishRespParamA	Allometric constant in standard respiration equation	*	30
fishRespParamB	Allometric exponent in standard respiration equation	none	0.784
fishRespParamC	Temperature coefficient in standard respiration equation	1/°C	0.0693
fishRespParamD	Velocity coefficient in activity respiration equation	s/cm	0.03

*This is an empirical parameter with units that depend on *fishRespParamB*.

4.3.8. Other energy losses

Many fish bioenergetic formulations include terms for energy losses due to egestion, excretion, and specific dynamic action. These terms are not included in *inSALMO* because their effects are small compared to the uncertainties and variability in food availability and in the feeding and growth formulation (Bartell et al. 1986). These terms may be important at extremely low or high temperatures when the ability to digest food can limit growth; instead, *inSALMO* uses the *cMax* function to limit food consumption at extreme temperatures.

4.3.9. Feeding strategy selection, net energy benefits, and growth

The feeding strategy selection, net energy, and growth methods calculate a fish's daily growth for a specific habitat cell. Total food and energy intake is calculated and total energy losses subtracted, determining whether drift feeding or active searching is more profitable.

Variables with the word “food” in their name refer to prey, in g; “energy” variables refer to energy from prey (j). Prey energy density (the habitat parameter *habPreyEnergyDensity*, j/g) is used to convert grams of prey eaten to joules of energy intake. Values of *habPreyEnergyDensity* are provided for various prey types by Hanson et al. (1997). A value of 2500 j/g is reasonable for streams where drift prey is dominated by aquatic insect larvae; a value of 4000 j/g is appropriate for streams where drift is dominated by higher-energy prey such as amphipods. Parameter *habPreyEnergyDensity* applies to both drift and search food.

The energy density of fish (fish parameter *fishEnergyDensity*, j/g) is used to convert a fish’s net energy intake to growth in weight. The energy density of salmonids actually varies through their life cycle (typically higher in adults, especially during gonad development prior to spawning), but this variation is ignored in *inSALMO*. The literature summarized by Hanson et al. (1997) indicates that 5900 j/g is a reasonable value for all stream salmonids.

The following steps describe the process used by a fish to determine the feeding strategy it would use, and the resulting food intake and growth it would obtain, for a particular habitat cell. This process uses variables (e.g., *driftIntake*, *feedTime*, *searchIntake*) calculated using the methods described above.

1. Determine the daily drift intake that would be obtained in the absence of more dominant fish in the cell. This *dailyPotentialDriftFood* (g/d) is determined from the hourly intake rates and hours spent feeding:

   dailyPotentialDriftFood = *driftIntake* × *feedTime*.

2. Determine *dailyAvailableDriftFood*, the drift intake rate available after more dominant fish in the cell have consumed their intake.

3. Calculate the actual drift intake rate *dailyDriftFoodIntake* (g/d), considering whether it is limited by actual food availability or the physiological maximum intake, *cMax*:

   dailyDriftFoodIntake = min(*dailyPotentialDriftFood, dailyAvailableDriftFood, cMax*).

4. Convert daily drift intake in grams of food to joules of energy, *dailyDriftEnergyIntake* (j/d):

   dailyDriftEnergyIntake = *dailyDriftFoodIntake* × *habPreyEnergyDensity*.

5. Conduct the bioenergetics energy balance to get net energy intake for drift feeding; total respiration (*respTotal*, j/d) depends on cell velocity and whether the fish has access to velocity shelter:

   dailyDriftNetEnergy = *dailyDriftEnergyIntake* - *respTotal*.

6. Determine the daily search feeding intake that would be obtained in the absence of more dominant fish in the cell, *dailyPotentialSearchFood* (g/d):

   dailyPotentialSearchFood = *searchIntake* × *feedTime*.

7. Determine *dailyAvailableSearchFood*, the search intake is available after more dominant fish have consumed their intake.

8. Calculate the actual search intake *dailySearchFoodIntake* (g/d), considering whether it is limited by food availability or maximum daily intake:

   dailySearchFoodIntake = min(*dailyPotentialSearchFood, dailyAvailableSearchFood, cMax*).

9. Convert daily search intake to joules of energy, *dailySearchEnergyIntake* (j/d):

   dailySearchEnergyIntake = *dailySearchFoodIntake* × *habPreyEnergyDensity*.

10. Conduct the bioenergetics energy balance to get net energy intake for search feeding:

    dailySearchNetEnergy = *dailySearchEnergyIntake* - *respTotal*.

11. Select the most profitable feeding strategy by comparing *dailyDriftNetEnergy* to *dailySearchNetEnergy*; and determine the energy intake for the best strategy:

    bestNetEnergy = max(*dailyDriftNetEnergy, dailySearchNetEnergy*).

12. Convert net energy intake to daily growth *dailyGrowth* (g/d):

    dailyGrowth = *bestNetEnergy* / *fishEnergyDensity*.

13. Update the fish's weight:

    fishWeight = *fishWeight* + *dailyGrowth*.

In the final step, *fishWeight* is not allowed to become negative; it is set to zero if *dailyGrowth* is negative with a magnitude greater than *fishWeight* (this can happen in the model, although biologically unrealistic, when small fish calculate growth for cells where swimming speed would be extremely high).

Examining how food intake and growth vary with cell velocity helps understand the feeding and growth formulation. Figure 14 illustrates how daily food intake (evaluated as the percent of *cMax*) varies with velocity, for both 5 cm juveniles and 15 cm salmonids, for both feeding strategies. Figure 15 illustrates the resulting growth (as percent body weight per day), also showing the effect of using velocity shelters on growth. These graphs assume the temperature is 15°, depth is 50 cm, feeding time is 16 h/d, *habShelterSpeedFrac* is 0.3, *habDriftConc* is 5×10^{-10}, and *habSearchProd* is 5×10^{-7} g/cm^2/h. Figure 16 is identical to Figure 15 except for depicting winter conditions, with a temperature of 5ºC and feeding time of 12 h.

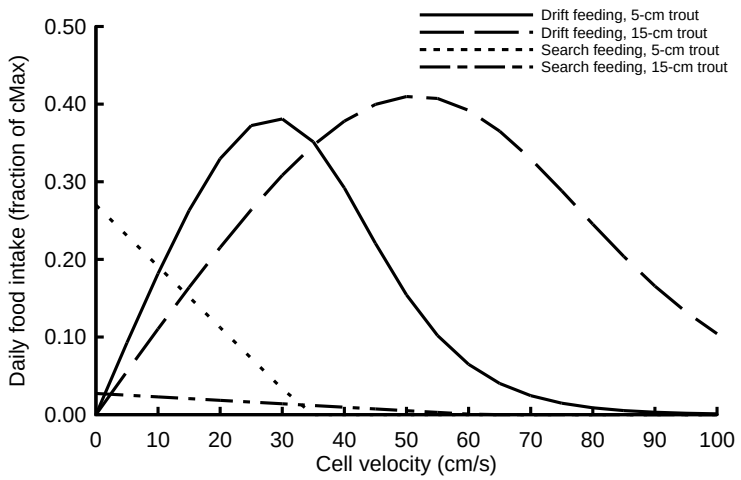


Figure 14. Variation in food intake with velocity for two sizes of salmonid, using drift or search feeding. Intake is depicted as percent of *cMax* (physiological maximum daily intake).

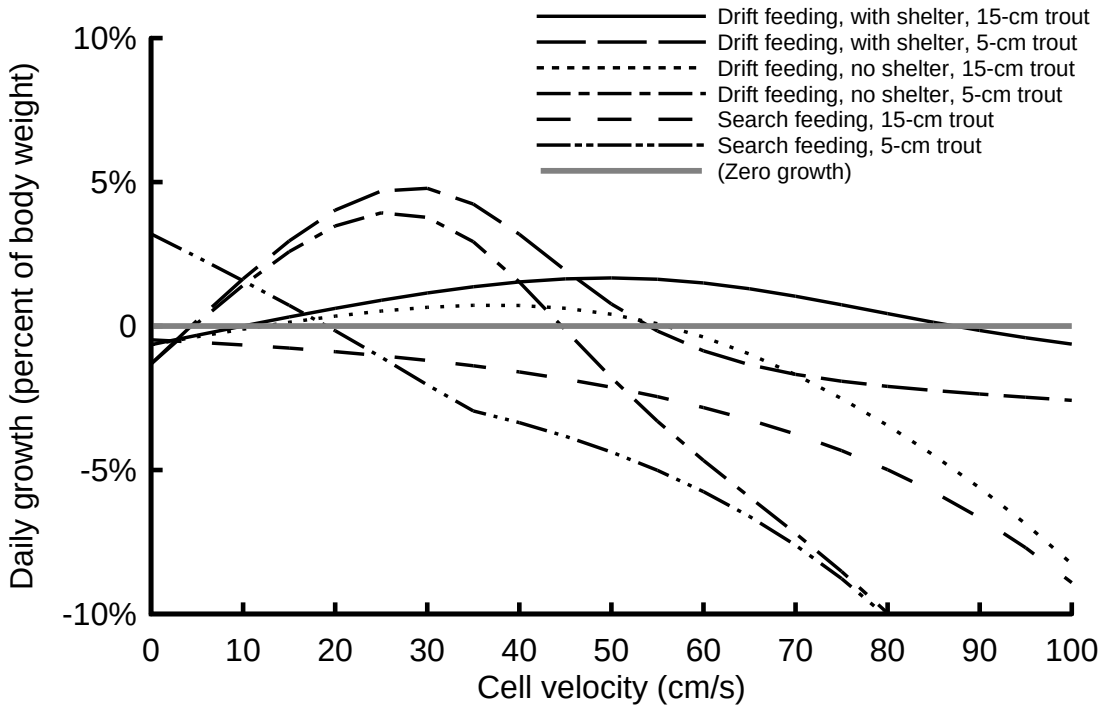


Figure 15. Variation in growth rate with velocity for two sizes of salmonid, drift and search feeding strategies. Growth is depicted as percent of body mass per day.

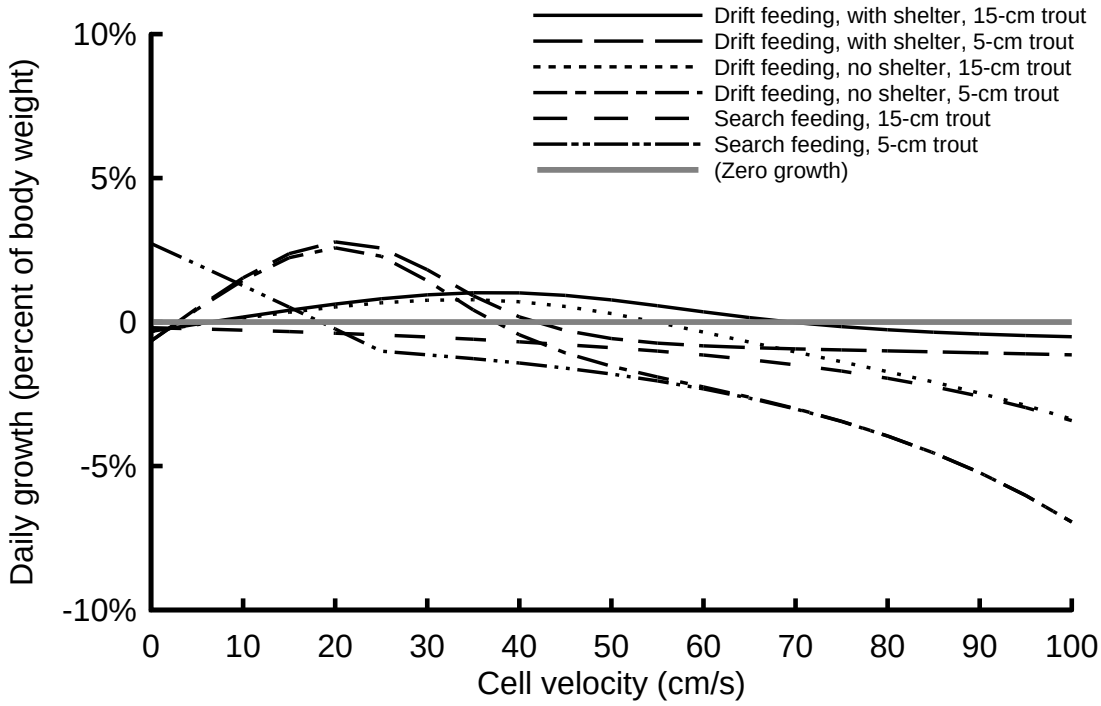


Figure 16. Variation in growth rate with velocity, under winter conditions.

Several patterns in these results are noteworthy in that they appear to reflect patterns observed in real salmon:

- Conditions providing high intake do not always provide high growth, due to the metabolic costs of swimming (especially for fish drift feeding without velocity shelters).
- The use of velocity shelters for drift feeding is very beneficial. Shelters increase the growth rate but also, more importantly, increase the range of velocities under which growth is positive.
- Larger fish can drift feed profitably over a wider range of velocities, and at higher velocities, than can smaller fish.
- Search feeding is a profitable strategy only for small fish in low velocities.
- The relative benefits of drift feeding increase with fish size.
- When temperatures are lower, growth is lower and optimized at lower velocities.

4.4. Fish Survival

Survival simulations determine, each day, which fish die from what causes. The survival action for a fish is a two-step process. First is calculating the probability of surviving each of several

mortality sources. Second is determining, stochastically, whether the fish actually dies due to any of the mortality sources.

The survival methods simulate important mortality sources: environmental and biological processes that can kill fish. Mortality sources are represented in *inSALMO* as survival probabilities: the daily probability of not being killed by one specific mortality source. The mortality sources in *inSALMO* are:

- High temperature,
- High velocity (exhaustion and inability to maintain position),
- Stranding (including predation risk associated with extremely shallow habitat),
- Poor condition (starvation and disease when weight is low),
- Predation by terrestrial animals, and
- Predation by fish.

The primary reason that *inSALMO* represents these different mortality sources separately is that the probability of surviving each varies differently with fish state and habitat conditions. For example, the risk of predation by terrestrial animals is greatest for large fish in shallow, low-velocity cells; the risk of predation by fish is greatest for small fish in deep cells. The primary adaptive behavior represented in *inSALMO*—habitat selection—depends on survival probabilities. For habitat selection to be modeled realistically, *inSALMO* must represent how different mortality sources vary differently over time, among fish, and over space. High temperature is included as a mortality source not as much because it affects habitat selection as because it is a way that river management can directly affect salmon.

Survival probabilities are used for two purposes. First, survival probabilities are used during habitat selection (Section 4.2) as a major input fish use in deciding which habitat cell to occupy. The second use, addressed here, is to model mortality: when and why each fish actually dies. The same methods are used to determine survival probabilities in modeling both habitat selection and mortality.

Death of fish is modeled stochastically by comparing pseudo-random numbers to the survival probabilities. Potential death due to each mortality source is treated as an independent event. On each simulated day, each fish determines whether it dies of each mortality source using these steps:

- Calculate the survival probability from the current state of the fish and its cell.
- Obtain a pseudo-random number from a uniform distribution between zero and one.
- If the random number is greater than the survival probability, then the fish dies as a result of the mortality source. No further mortality sources are evaluated for the fish.

- If the fish does not die, then the next mortality source is evaluated.

While death due to each mortality source is treated independently, the order in which mortality sources are evaluated can have a (usually very small) effect on how many fish die of each kind of mortality. The ordering of mortality sources is discussed with the model schedule in Section 8.2.

It is important to understand that seemingly high daily survival probabilities can result in low survival over time. For example, a daily survival probability of 0.99 results in mortality of 26 percent of fish within 30 days ($0.99^{30} = 0.74$). Survival probabilities should be well above 0.99 if they are not to cause substantial mortality over time. It is often helpful to translate daily survival values into the probability of surviving for 30 days and think about monthly survival.

The following sections describe the detailed formulation used to calculate survival probabilities for each mortality source.

4.4.1. High temperature

This mortality source represents the breakdown of physiological processes at high temperatures. It does not represent the effect of high temperatures on bioenergetics (reduced growth at high temperature). The high temperature survival function is based on laboratory data collected from (presumably) disease-free fish, so it 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.

While input to *inSALMO* includes only daily mean temperature, mortality is related to the daily maximum temperature as well as the mean (although the relative importance of mean v. maximum temperature is not clear: Dickerson and Vinyard 1999, Hokanson et al. 1977). The survival probability parameters therefore assume a difference between mean and peak temperatures. 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 mortality varies with laboratory conditions and techniques and the endpoints used to define mortality; varies between laboratory and field conditions; and undoubtedly varies among individuals. Review of such literature compilations as Behnke (1992) and Moyle and Marchetti (unpublished) indicates that any differences in measured lethal temperatures among salmonid species are not clearly distinguishable from uncertainty and variability in the measurements. Recent laboratory data showed approximately 60 percent survival of golden trout (*Oncorhynchus mykiss*) juveniles over a 30-d period at a constant 24° (Myrick 1998), 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 parameters in Table 10 (illustrated in Figure 17) were derived from information compiled by Myrick and Cech (2004); they produce survival survival of 0.98 at 24°, 0.88 at 26°, and < 0.5 at 28°.

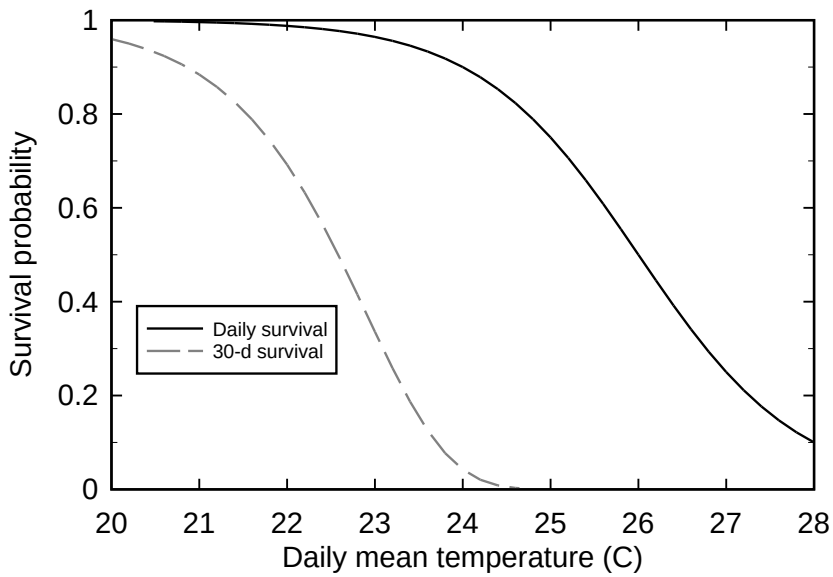


Figure 17. Survival probability function for high temperature. Daily survival (solid line) is the probability of a salmon surviving high temperature mortality for one day. 30-d survival (dashed line) is the probability for surviving the temperature for 30 days (equal to the daily survival raised to the power 30).

Table 10. Parameter values for high temperature mortality.

Parameter	Definition	Value
mortFishHiTT9	Daily mean temperature (°C) at which high temperature survival is 90 pct	24
mortFishHiTT1	Daily mean temperature (°C) at which high temperature survival is 10 pct	28

4.4.2. High velocity

The high velocity survival function represents the potential for salmon to suffer fatigue or lose their ability to hold position in a cell with high velocity. This function is included not because salmon often die due to high velocity, but because it strongly affects habitat selection: mortality due to high velocities is not observed in nature because fish avoid it by moving. Velocities posing mortality risk can be widespread at high flows, but can also occur (especially for small fish) at normal flows.

The survival probability is based on the ratio of the swimming speed a fish uses in a cell to the fish's maximum sustainable swim speed. The swimming speed used in a cell is determined when calculating respiration energy costs (Section 4.3.7): fish are assumed to swim at the cell's water velocity unless they are drift-feeding with access to velocity shelters. Fish using velocity shelters are assumed to swim at a speed equal to the cell's velocity times the parameter *habShelterSpeedFrac*.

Maximum sustainable swim speed (*maxSwimSpeed*, cm/s) is a particularly important state variable for model salmon. As a component of both high velocity mortality and drift feeding (Section 4.3.3), *maxSwimSpeed* strongly affects the relationship between a cell's velocity and habitat quality for various size salmon. Because *inSALMO* uses a daily time step, the maximum swim speed used for high velocity mortality must be a speed that fish can swim for hours, not a burst or short-term maximum speed. The formulation for *maxSwimSpeed* is based on literature values of "critical swimming speed" (often abbreviated as 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. To model *maxSwimSpeed*, long-term values of U_{crit} were used. Myrick (1998) cites references indicating that salmonids 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 stream salmonids 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.]

There is considerable variability among these studies, likely due 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, *maxSwimSpeed* increases with fish length (Figure 18). Second, *maxSwimSpeed* varies nonlinearly with temperature, peaking at temperatures around 10-15º (Figure 19). The formulation for *maxSwimSpeed* therefore has two terms: the first represents how swimming speed at 10-15º varies with fish length, and the second modifies *maxSwimSpeed* for temperature.

$$maxSwimSpeed = [(fishMaxSwimParamA \times fishLength) + fishMaxSwimParamB] \times [(fishMaxSwimParamC \times temperature^2) + (fishMaxSwimParamD \times temperature) + fishMaxSwimParamE]$$

Parameter values are in Table 11. These parameters were fit to data from the studies cited above. Observations of U_{crit} from these studies were converted to maximum sustainable swimming speeds by multiplying U_{crit} by 0.9. The relation between *maxSwimSpeed* and salmon length (parameters *fishMaxSwimParamA* and *fishMaxSwimParamB*) was fit using observations made at temperatures between 10 and 15ºC (Figure 18). A few of these literature values were omitted as outliers (as shown in the figures) because they appeared to underestimate swim speed. Parameters *fishMaxSwimParamC*, *fishMaxSwimParamD*, and *fishMaxSwimParamF* were fit via polynomial regression of (a) the ratio of swim speed at a temperature to swim speed at 15º in the same study, vs. (b) temperature (Figure 19).

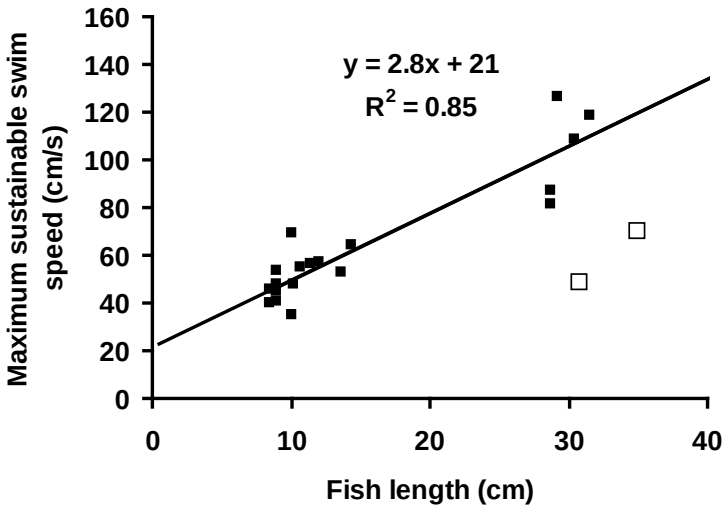


Figure 18. Maximum sustainable swimming speed as a function of fish length; measurements made at 10-15º C. The points marked as open squares were omitted as outliers.

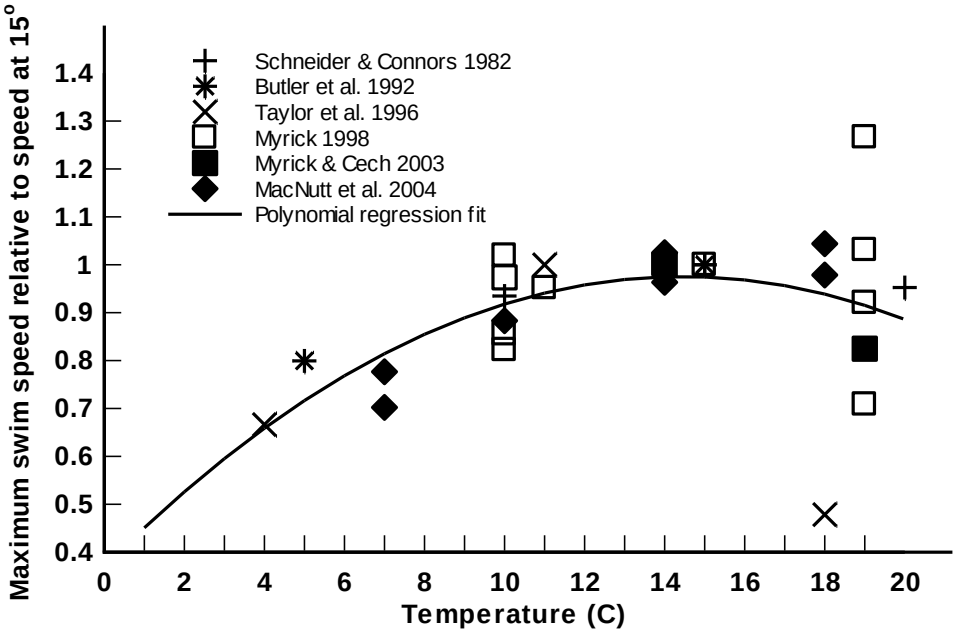


Figure 19. Variation in maximum sustainable swim speed with temperature. Observations from four studies are shown separately. The Y axis is the measured swim speed divided by the speed measured at (or near) 15º in the same study.

A decreasing logistic function relates survival probability to the fish's swimming speed in its habitat cell divided by the fish's value of *maxSwimSpeed* (Figure 20). The parameters for this function (Table 11) are chosen so that high velocity mortality is negligible at swimming speeds less than *maxSwimSpeed*, reflecting that (a) the laboratory equipment for measuring swim speeds does not provide the kinds of turbulence and fine-scale velocity breaks that salmon can often use to reduce swimming effort in natural conditions, and (b) stream fish are likely to be in better condition than laboratory fish.

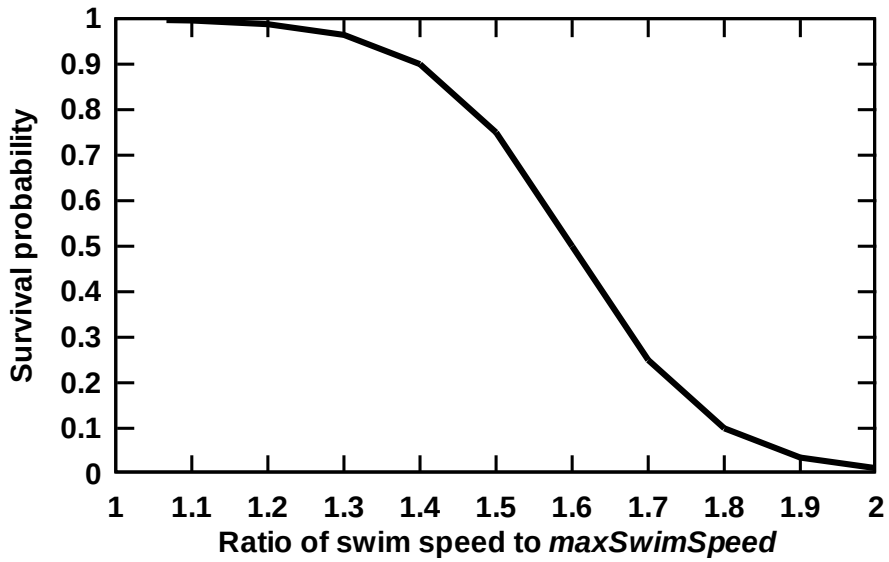


Figure 20. Survival probability function for high velocity. The X axis is the fish's actual swimming speed divided by its maximum sustainable swimming speed.

Table 11. Parameter values for high velocity mortality.

Parameter	Definition	Value
fishMaxSwimParamA	Length coefficient in maximum swim speed equation (1/s)	2.8
fishMaxSwimParamB	Constant in maximum swim speed length term (cm/s)	21
fishMaxSwimParamC	Temperature squared coefficient in maximum swim speed equation ($^oC^{-2}$)	-0.0029
fishMaxSwimParamD	Temperature coefficient in maximum swim speed equation ($^oC^{-1}$)	0.084
fishMaxSwimParamE	Constant in maximum swim speed temperature term (unitless)	0.37
mortFishVelocityV9	Ratio of fish swimming speed to maximum swim speed at which high velocity survival is 90 pct (unitless)	1.4
mortFishVelocityV1	Ratio of fish swimming speed to maximum swim speed at which high velocity survival is 10 pct (unitless)	1.8

4.4.3. Stranding

Stranding mortality represents the death of fish that are unable to move out of cells that become extremely shallow or dry as flow decreases. Fish in *inSALMO* already have a strong incentive to avoid cells with near-zero depth: drift food intake and survival of terrestrial predation are low. However, there can be cases where (a) a fish is limited by its maximum movement distance from reaching a cell with non-zero depth, or (b) no better habitat is available for other reasons.

Survival of stranding is modeled as an increasing logistic function of depth divided by fish length (Figure 21; Table 12). Because the terrestrial predation function does not represent the greatly increased likelihood of predation when depth is extremely low (e.g., when fish are trapped in isolated pools; Harvey and Stewart 1991), this risk is included as part of stranding mortality. The stranding survival function does not distinguish whether fish in very low or zero depths die from lack of water or from predation.

The stranding parameters do not cause survival to reach zero when depth is zero, reflecting that real habitat (as opposed to the model's cells) has variation in bottom elevation- some water could remain even if a cell's simulated depth becomes zero. Depth is divided by fish length to scale how the risks of low depths vary with fish size: shallow habitat that may be very valuable for small fish (protecting them from aquatic predation) may pose a stranding risk for large fish.

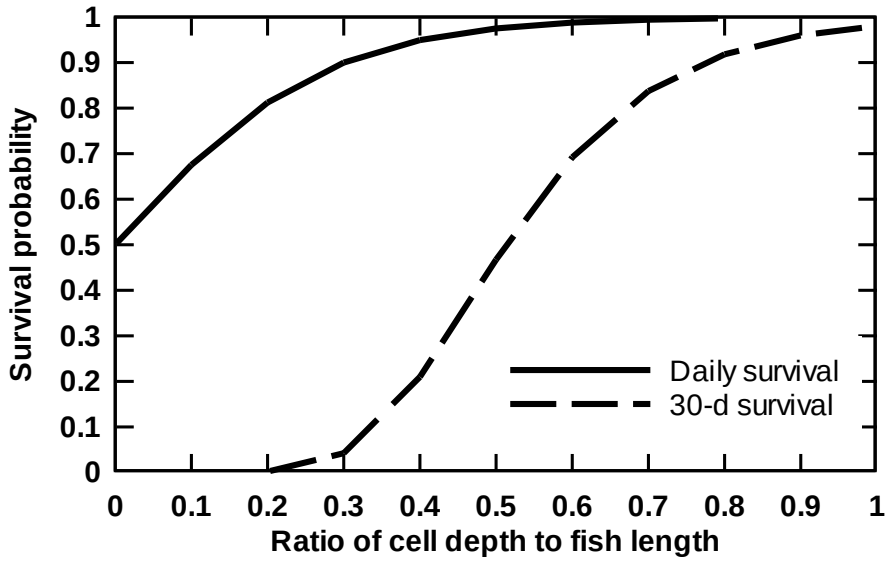


Figure 21. Survival probability function for stranding, showing the probability for surviving one day (solid line) and for 30 days (dashed line).

Table 12. Parameter values for stranding mortality.

Parameter	Definition	Units	Value
mortFishStrandD1	Ratio of depth to fish length at which stranding survival is 10 pct	none	-0.3
mortFishStrandD9	Ratio of depth at which stranding survival is 90 pct	none	0.3

4.4.4. Poor condition

Fish in poor condition (low value of the condition factor *K*, weight in relation to length; Section 4.3.1) are at risk of starvation, disease, and excess vulnerability to predators. These risks are combined in the poor condition survival probability. 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 intake;
- Survival is lower at higher swimming activity and temperature (which both increase metabolism);
- Relative weight (equivalent to *K*) decreased linearly over time during starvation; but
- Mortality was predicted better by an index of lipid content than by *K*; one reason is that lipids are replaced by water as energy stores are depleted.

Unfortunately, modeling how body lipids are depleted and replaced by water and related processes would add considerable complexity and uncertainty to *inSALMO*, as they are not well understood. Instead, poor condition survival probability is represented as an increasing logistic function of *K* with parameter values estimated to provide reasonable survival probabilities over several days and weeks (Figure 22; Table 13). The parameters produce a survival probability less than 100 percent even when *K* is at its maximum of 1.0, because disease can occur (though is less likely) when condition is relatively good.

Poor condition is a unique mortality source in that fish can never increase their survival probability immediately by selecting different habitat. Fish in poor condition have a strong incentive to select habitat that provides rapid growth so their condition increases; however, sufficient growth to recover high condition takes a number of days. Even apparently high daily survival probabilities for this mortality source (e.g., 0.90) result in a low probability of surviving until normal weight can be regained. As Figure 22 indicates, the probability of surviving for extended periods becomes quite low when *K* falls below 0.8.

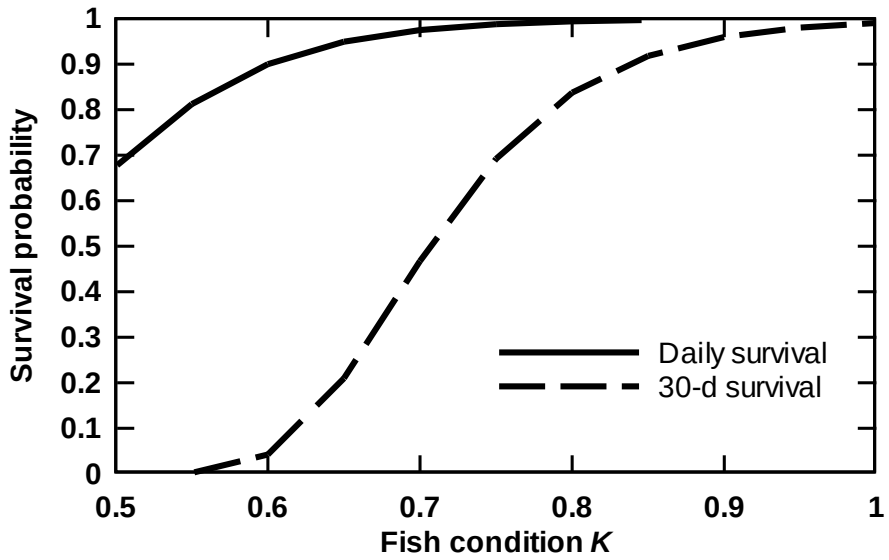


Figure 22. Survival probability function for poor condition. The dotted line is the probability for surviving for 30 d at the value of *K*.

Table 13. Parameter values for poor condition mortality.

Parameter	Definition	Value
mortFishConditionK1	Fish condition factor *K* at which survival is 10 pct (unitless)	0.3
mortFishConditionK9	*K* at which survival is 90 pct (unitless)	0.6

Before modifying the parameters for poor condition, users of *inSALMO* should be aware that poor condition mortality can have a strong effect on habitat selection (Section 4.2.2) as well as mortality. As a consequence, changes in parameter values are likely to have widespread, complex, and unexpected effects. For example, one might assume that increasing the survival probability (e.g., by decreasing *mortFishConditionK9* from 0.6 to 0.7) would result in less mortality due to poor condition. However, because fish select habitat using a tradeoff between poor condition and other (primarily, predation) mortality sources, this change in parameters could result in fish selecting different habitat that has lower growth and lower predation risk, at least partially offsetting the expected reduction in poor condition mortality.

4.4.5.Terrestrial predation

Predation by terrestrial animals is a dominant source of mortality to salmon, especially adults (Alexander 1979, Harvey and Marti 1993, Metcalfe et al. 1999, Quinn and Buck 2001, Valdimarsson et al. 1997). The terrestrial predation formulation represents predation by a mix of such predators as otters, raccoons, snakes, herons, mergansers, kingfishers, and dippers.

Characteristics of terrestrial predators that affect the survival probability function include they generally (but not always):

- Are bigger than salmon,
- Are poorer swimmers than relatively large salmonids,
- Are warm-blooded, and
- Locate fish prey from the air.

These characteristics vary among predators, but they lead to these generalizations about terrestrial predation:

- Big salmon are vulnerable, often more vulnerable than very small salmon;
- Risks are year-round because warm-blooded predators feed as much or more in winter (except those that hibernate or migrate); and
- Salmon are more at risk when more visible from the air.

Results of the *inSTREAM* individual-based trout model, which uses the same formulation as *inSALMO* for predation survival, are quite sensitive to how terrestrial predation varies with habitat variables such as depth and velocity. Results of *inSALMO* are expected to be less sensitive to terrestrial predation because (a) adult salmon are only present for short times, so predation risk does not control their abundance, and (b) juveniles are still vulnerable to terrestrial predation but likely much more vulnerable to predation by other fish—at least at sites where where non-salmonid predators such as pikeminnow and bass are abundant.

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 between zero and one, with higher values for greater protection from predation. The survival increase functions are assumed to act independently. Therefore, the terrestrial predation survival probability (terrPredSurv) is obtained by increasing the minimum survival (decreasing the difference between minimum survival and 1.0) by the <u>maximum</u> of the independent survival increase functions. This assumption is expressed mathematically as:

$$terrPredSurv = mortFishTerrPredMin + [(1 - mortFishTerrPredMin) \times \max(terrPredDepthF, terrPredTurbidityF, terrPredLengthF \ldots)].$$

where *terrPredDepthF*, *terrPredTurbidityF*, etc. are the values of the survival increase functions described below.

Using this approach, the value of *terrPredSurv* does not vary with how many survival increase functions there are, but instead is only limited by one function at a time. Survival increase functions can be added, removed, or revised without re-calibrating the overall predation survival rate. However, the approach does not represent the potential combined effects of, for example,

using deeper and faster habitat. Both depth and velocity make fish more difficult to see, and the combination of deep and fast is safer than only deep or fast; but this combined effect is not represented in this formulation.

The value of *mortFishTerrPredMin* is assumed to be the daily probability of surviving terrestrial predation under conditions where the survival increase functions are minimal (offering no reduction in risk). Field data for estimating this minimum survival are unlikely to be available, so it is best estimated by calibrating the model to observed abundance and habitat use patterns.

The following survival increase functions are included. (The effect of any function can be turned off by setting its function's parameters to yield values near zero.) Suggested parameter values are provided at the end of the section (Table 14).

Depth. Fish are more vulnerable to terrestrial predators when in shallow water, where they are easier for predators to locate and catch. The depth survival increase function is an increasing logistic curve: survival increases as depth increases (Figure 23). 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. (Note that the very high risk of terrestrial predation that occurs when fish are in near-zero depths is included in stranding mortality.)

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. Figure 23 illustrates parameter values for small streams and large rivers (Table 14).

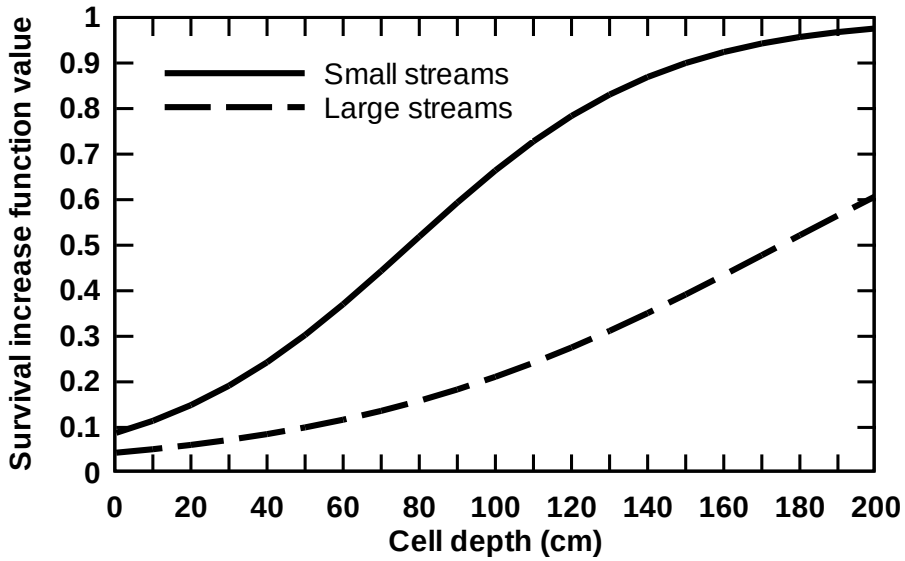


Figure 23. Depth survival increase function for terrestrial predation survival.

Turbidity. Turbidity makes fish less visible to terrestrial predators and, because detection from the air is key to terrestrial predation success, is assumed to be an important survival increase function. No literature directly relating terrestrial predation to turbidity was found. Instead, this formulation considers the observed effect of turbidity on the ability of fish to detect prey (Section 4.3.3), 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 lengths of water than must fish predators. Therefore, the turbidity survival increase function has little effect at values below 5 NTUs but strongly reduces terrestrial predation risk at >40 NTUs (Figure 24).

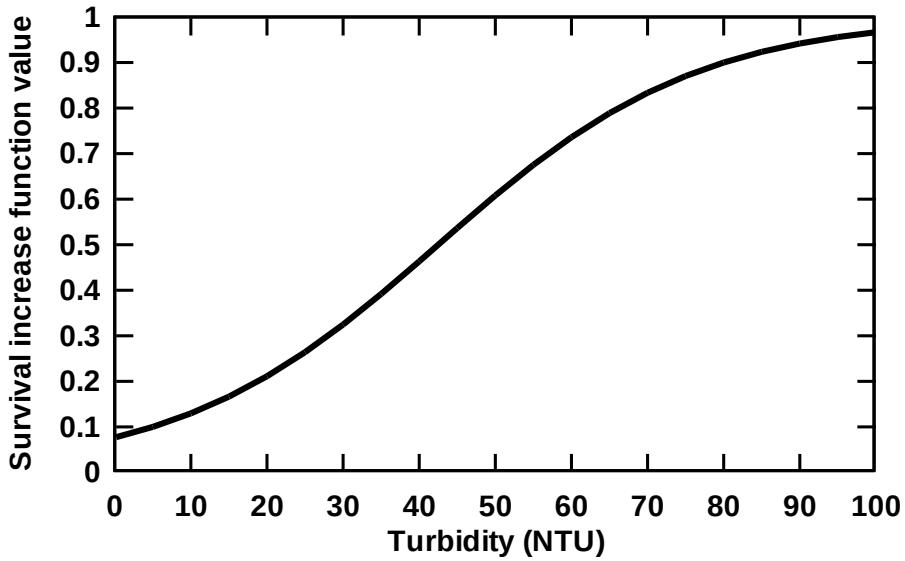


Figure 24. Turbidity survival function for terrestrial predation survival.

Fish length. Small fish are less vulnerable to terrestrial predation, presumably 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. Dippers (*Cinclus mexicanus*) are an example terrestrial predator that selects salmonid fry and other small fish (Thut 1970), so very small fish are not invulnerable to terrestrial predation. Therefore, survival of terrestrial predation is assumed to decrease with fish length, but only fish less than 4 cm in length are relatively protected (Figure 25). These parameter values should be reconsidered for sites where predation is dominated by larger mammals (otters, bears) that strongly prefer large fish.

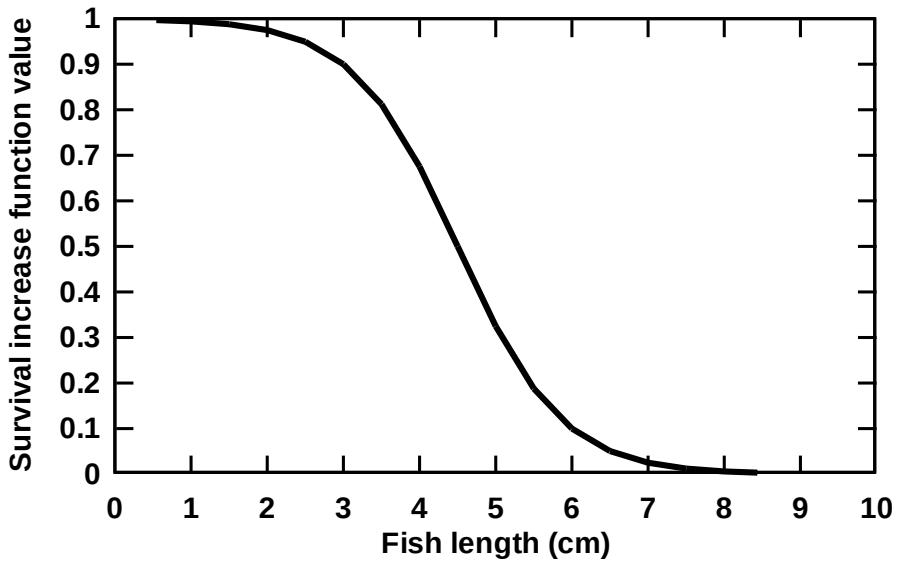


Figure 25. Fish length survival increase function for terrestrial predation survival.

Feeding time. Fish are much more vulnerable to predation when they are actively feeding during the day instead of resting and hiding at night (Metcalfe et al. 1999). The survival increase function is modeled as a decreasing function of *feedTime* (h), the hours spent feeding per day (Section 4.3.2). Parameters are chosen so survival decreases nearly linearly with feedTime (Figure 26).

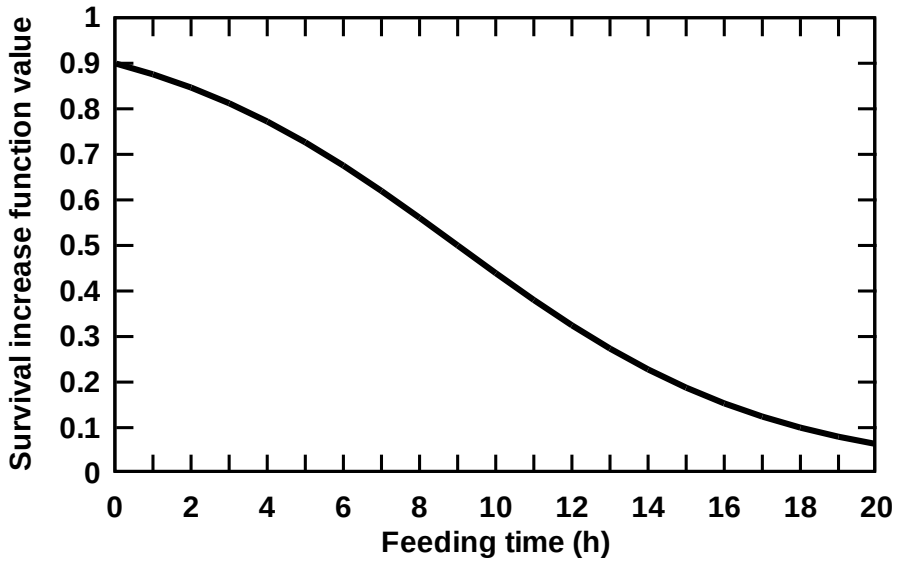


Figure 26. Feeding time function for terrestrial predation mortality.

Water velocity. Water velocity is assumed capable of increasing terrestrial predation survival because (1) velocity-caused turbulence makes fish harder to see, and (2) some predators are poorer swimmers than salmon so they are expected to be less able capture fish in faster water. The survival increase function is therefore an increasing logistic curve that provides sharply increasing protection from terrestrial predators at velocities above 50 cm/s (Figure 27). As with the depth survival increase function, useful parameter values for the velocity function may differ between small and large streams. In small streams, high velocities combine with high turbulence and obstacles to make swimming difficult. In large rivers, however, there can be run habitat where velocities are high while turbulence is low, so good swimmers such as mergansers and otters may perform quite well. Two sets of parameter values are provided in Table 14 and illustrated in Figure 27.

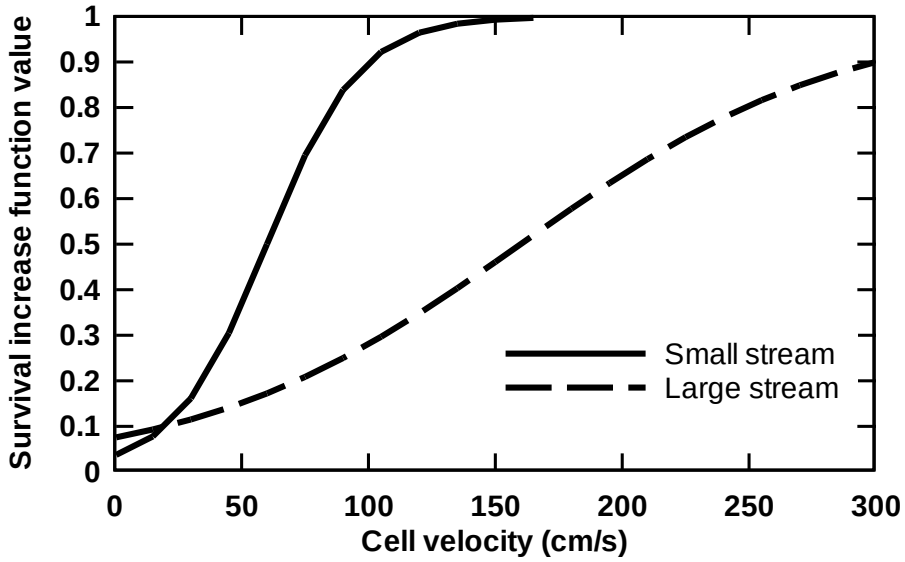


Figure 27. Velocity survival increase function for terrestrial predation survival, with parameters for both small streams and large rivers.

Temperature. No temperature-based survival increase function is included in *inSALMO* because there are no clear mechanisms that would cause terrestrial predation pressure (unlike fish predation) to change with temperature. There is not a good basis for assuming predator activity is lower in winter; most important terrestrial predators are warm-blooded and many do not hibernate. In fact, such predators need additional food to maintain their metabolic needs in winter. The reduced swimming ability of salmon at low temperatures can also offset any decreased activity by predators by reducing the ability of salmon to escape (Metcalfe et al. 1999). Terrestrial predation can be greatly reduced when rivers freeze over, but ice is not represented in *inSALMO*.

Distance to hiding cover. Fish can avoid mortality by hiding when predators are detected. The success of this tactic depends on the presence of hiding cover and the distance the fish must travel to reach it. The value of hiding cover is one habitat function that clearly occurs at a spatial scale different from the cell size typically used in *inSALMO*; hiding cover several to tens of meters from a fish can provide at least some predation protection.

Hiding cover is represented with a survival increase function that increases as distance to hiding cover decreases. Distance to cover (*cellDistanceToHide*, cm) is an input for each habitat cell, estimated in the field as the average distance a fish in the cell would need to move to hide from a predator. The value of *cellDistanceToHide* 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 hiding cover (< 100 cm) 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 salmon from hiding (e.g., otters). Cover several meters away is still valuable for escaping from terrestrial predators that have been detected. Therefore, the effect of distance to hiding cover is modeled as a decreasing logistic function of *cellDistanceToHide* (Figure 28).

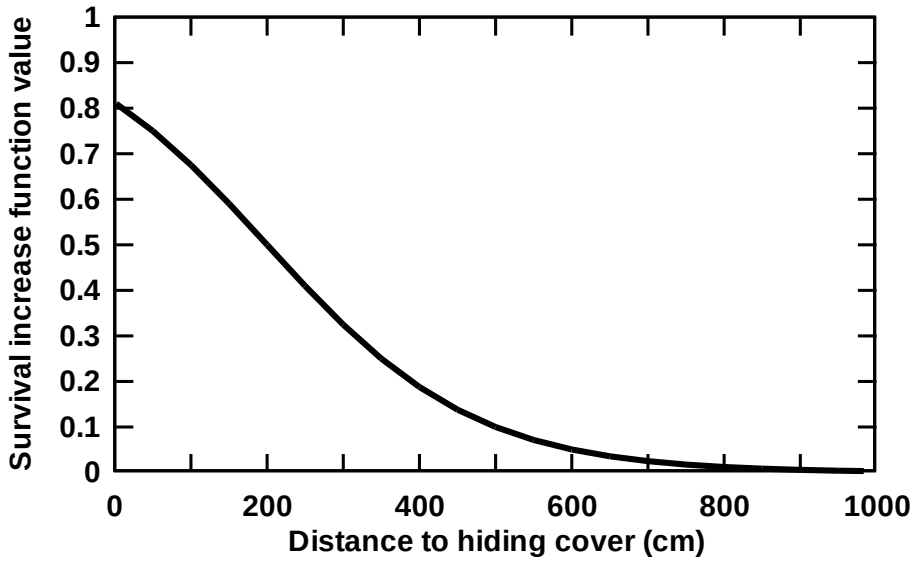


Figure 28. Distance to hiding cover function for terrestrial predation survival.

Table 14. Parameter values for terrestrial predation mortality. Values are estimates for mid-sized spawning streams and should be re-considered for each site.

Parameter	Definition	Value
mortFishTerrPredMin	Daily survival probability due to terrestrial predators under most vulnerable conditions (unitless)	0.99 (until fit via calibration)
mortFishTerrPredD1	Depth at which survival increase function is 10 pct of maximum (cm)	5
mortFishTerrPredD9	Depth at which survival increase function is 90 pct of maximum (cm)	200
mortFishTerrPredL9	Fish length at which survival increase function is 90 pct of maximum (cm)	3
mortFishTerrPredL1	Fish length at which survival increase function is 10 pct of maximum (cm)	6
mortFishTerrPredF9	Feeding time at which survival increase function is 90 pct of maximum (h)	0
mortFishTerrPredF1	Feeding time at which survival increase function is 10 pct of maximum (h)	18
mortFishTerrPredV1	Velocity at which survival increase function is 10 pct of maximum (cm/s)	20
mortFishTerrPredV9	Velocity at which survival increase function is 90 pct of maximum (cm/s)	200
mortFishTerrPredH9	Distance to hiding cover at which survival increase function is 90 pct of maximum (cm)	-100
mortFishTerrPredH1	Distance to hiding cover at which survival increase function is 10 pct of maximum (cm)	500
mortFishTerrPredT1	Turbidity at which survival increase function is 10 pct of maximum	10
mortFishTerrPredT9	Turbidity at which survival increase function is 90 pct of maximum	50

4.4.6. Aquatic predation

The aquatic predation formulation represents mortality due to predation by fish. This formulation has been modified from that of *inSTREAM* (Railsback et al. 2009) by removing the assumption that the dominant source of aquatic predation is cannibalism. Instead, inSALMO assumes that juvenile salmon are vulnerable to predation by an unspecified community of predatory fish and that the predation risk is unrelated to the simulated salmon population.

As with terrestrial predation, the formulation uses a minimum survival probability that applies when fish are most vulnerable to aquatic predation, and a number of survival increase functions.

aqPredSurv = *mortFishAqPredMin* +
[(1–*mortFishAqPredMin)* × max*(aqPredDepthF, aqPredLengthF, aqPredVelF* ...)]

where *aqPredSurv* is the daily survival probability for a particular fish in a particular habitat cell and *aqPredDepthF*, *aqPredLengthF*, etc. are survival increase function values. The value of *mortFishAqPredMin* is the daily probability of surviving aquatic predation under conditions where the survival increase functions offer no reduction in risk. As with terrestrial predation, data for directly estimating aquatic risks are unlikely to be available, so it is recommended that *mortFishAqPredMin* be estimated by calibrating the model to observed patterns of abundance and habitat selection by juvenile fish.

The aquatic predation survival formulation includes the following functions for survival increase functions. Parameter values are given at the end of the section (Table 15).

Distance to hiding cover. Salmon juveniles are assumed to use hiding cover to avoid fish as well as terrestrial predators, so a survival increase function for cell distance to hiding cover is included. (This is a change from the trout model formulation of Railsback et al. 2009.) The hiding cover formulation has the same form as that for terrestrial predation, a decreasing logistic function of *cellDistanceToHide* (Figure 28). Separate parameters (*mortFishAqPredH9*, *mortFishAqPredH1*) are used to define the hiding cover function for aquatic predation, so it can have a different shape than that for terrestrial predation. The hiding cover parameters for aquatic predation were chosen to represent the assumptions that, for small salmon, hiding cover must be quite close to provide protection from fish, and that even dense cover does not provide complete protection from smaller predators such as trout.

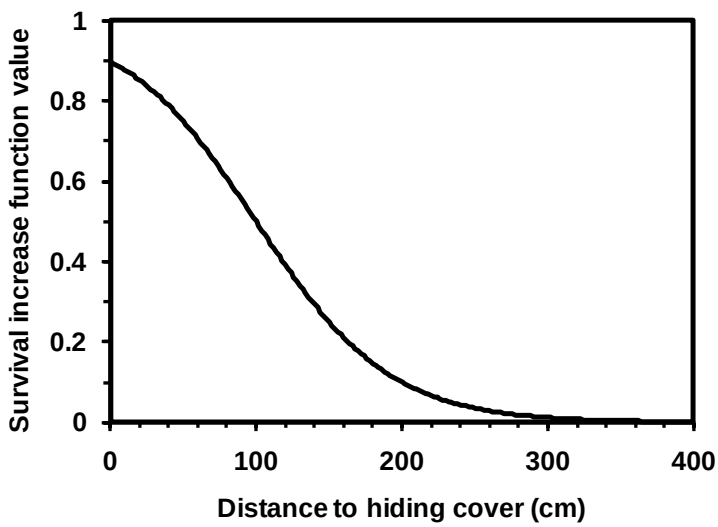


Figure 29. Hiding cover survival increase function for aquatic predation.

Depth. Aquatic predation survival is assumed to be high in water shallow enough to physically exclude large fish, or shallow enough to place large fish at high risk of terrestrial predation. The depth survival increase function is therefore a decreasing logistic function, with high survival at depths less than 5 cm (Figure 30).

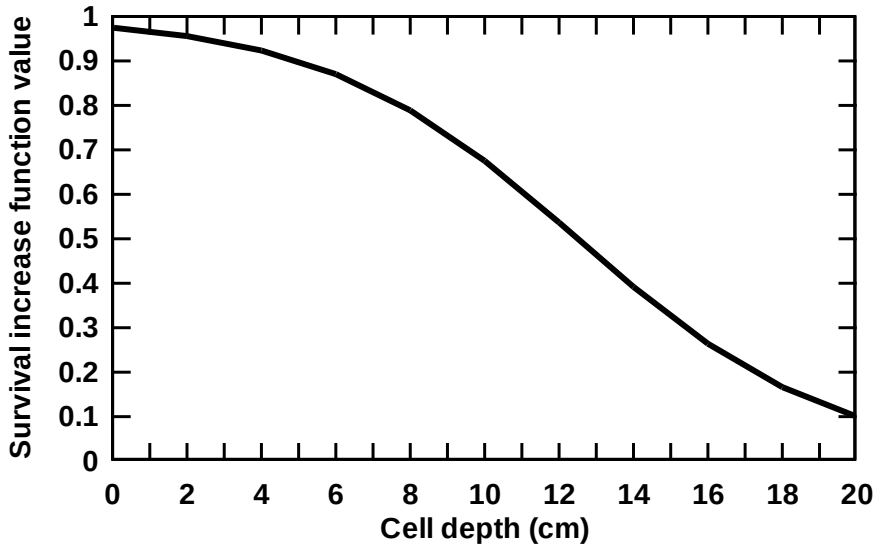


Figure 30. Depth survival increase function for aquatic predation survival.

Fish length. As fish grow, they become better able to out-swim piscivorous fish 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 31 illustrates parameters for sites where the predator fish community includes large-gaped piscivores such as pikeminnow and bass.

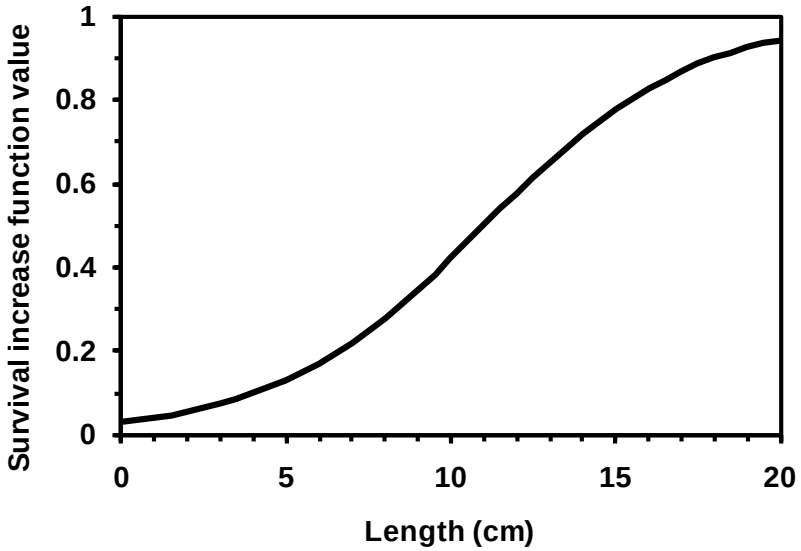


Figure 31. Fish length survival increase function for aquatic predation survival.

Feeding time. This survival increase function is the same for aquatic predation as it is for terrestrial predation. The survival increase is a decreasing logistic function of *feedTime*, the number of hours per day spent foraging. Separate parameters control the feeding time function for aquatic vs. terrestrial predation, but the values recommended above for terrestrial predation are also recommended for aquatic predation.

Low temperature. This survival increase function reflects how low temperatures reduce the metabolic demands, swimming ability, and, therefore, feeding activity of piscivorous fish. The function is a decreasing logistic function (Figure 32) that approximates the decline in metabolism and feeding with temperature of a mixed community of warmwater (e.g., pikeminnow) and coldwater (e.g., trout) predators.

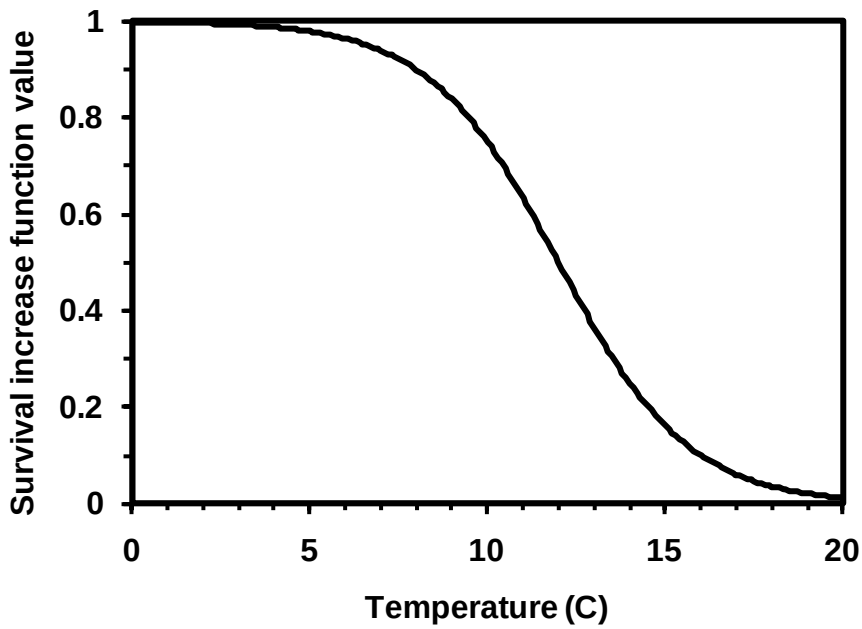


Figure 32. Temperature survival increase function for aquatic predation survival.

Turbidity. The survival increase function for turbidity represents how encounter rates between predator and prey fish decline as turbidity increases. The turbidity function is based on experimental observations and citations provided by Gregory and Levings (1999). Gregory and Levings compared piscivory by fish in adjacent clear and turbid rivers and found piscivory much lower, but still present, in the turbid river. Turbidity appears to reduce the ability of piscivorous fish to detect prey fish and thus the encounter rate between predator and prey (Gregory and Levings 1999, Vogel and Beauchamp 1999, DeRobertis et al. 2003). One 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 where small fish are likely to be found (Vogel and Beauchamp 1999). The parameters for this function provide no protection from aquatic predation at low turbidities and a 50 percent reduction in risk at 40 NTU (Figure 33). As turbidity continues to increase toward extreme values, aquatic predation risk continues to decrease but is not eliminated.

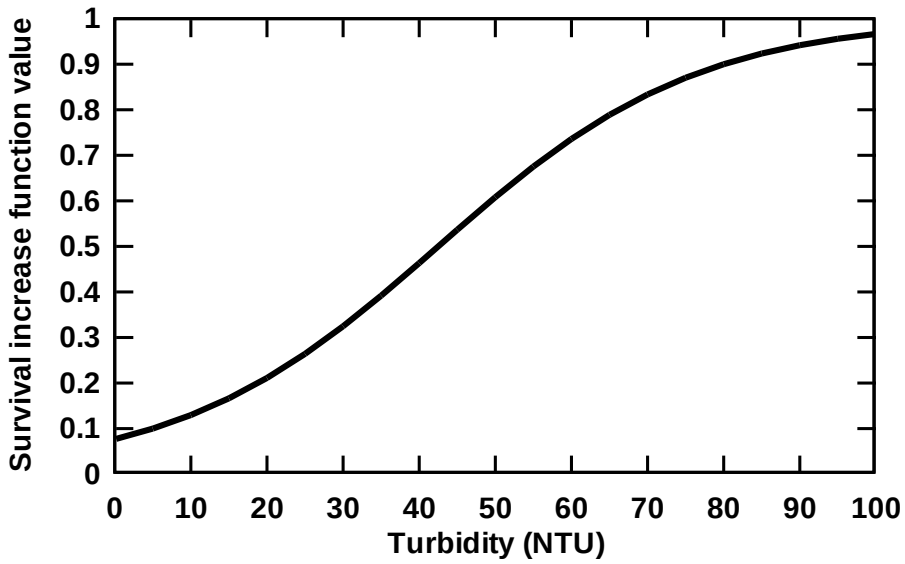


Figure 33. Turbidity survival increase function for aquatic predation survival.

Table 15. Parameter values for aquatic predation mortality. Parameter values are for sites where large-gaped, warm- and coldwater predators are abundant.

Parameter	Definition	Value
mortFishAqPredMin	Daily survival probability due to aquatic predators under most vulnerable conditions (unitless)	0.9 (until fit via calibration)
mortFishAqPredH9	Distance to hiding cover at which survival increase function is 90 pct of maximum (cm)	0
mortFishAqPredH1	Distance to hiding cover at which survival increase function is 10 pct of maximum (cm)	200
mortFishAqPredD9	Depth at which survival increase function is 90 pct of maximum (cm)	10
mortFishAqPredD1	Depth at which survival increase function is 10 pct of maximum (cm)	20
mortFishAqPredL1	Fish length at which survival increase function is 10 pct of maximum (cm)	4
mortFishAqPredL9	Fish length at which survival increase function is 90 pct of maximum (cm)	18
mortFishAqPredF9	Feeding time at which survival increase function is 90 pct of maximum (h)	0
mortFishAqPredF1	Feeding time at which survival increase function is 10 pct of maximum (h)	18
mortFishAqPredT9	Temperature at which survival increase function is 90 pct of maximum (°C)	8
mortFishAqPredT1	Temperature at which survival increase function is 10 pct of maximum (°C)	16
mortFishAqPredU9	Turbidity at which survival increase function is 90 pct of maximum (NTU)	80
mortFishAqPredU1	Turbidity at which survival increase function is 10 pct of maximum (NTU)	5

4.4.7. Demonic intrusion: experimenter-induced mortality

The graphical interface of *inSALMO*'s software allows the user to select individual salmon and remove them from the simulation. This capability can be useful for conducting controlled simulation experiments (e.g., Railsback and Harvey 2002 used it to look at how a hierarchy of adult trout shifted as the largest individuals were removed). Fish that are killed by the experimenter in this way are labeled as having died of "demonic intrusion", a term used by Hurlbert (1984) for the effects of experimenters on their study systems. There are no

parameters or equations related to demonic intrusion mortality, but it appears as a potential mortality source in model output.

5. Redds

Redds are the nests laid by spawning salmon. In *inSALMO*, a redd and the eggs it contains are modeled as one object: individual fish are not tracked until they emerge. The model redds keep track of the number of eggs remaining alive and determine when the eggs turn into new salmon. The species of a redd and its initial number of eggs are determined by the female spawner that created the redd (Section 4.1.3).

Because of its objectives as a management model, *inSALMO* models redds with relatively little biological detail but with substantial detail in how stream flow and temperature affect egg incubation and survival. The following are among the processes that can affect salmonid spawning success (see, e.g., Groot and Margolis 1991) that are not considered explicitly in *inSALMO*.

- Eggs can be killed by a variety of predators and parasites.
- Gravel size, fine sediment, and water quality can affect egg survival and development rates. In particular, low flow of water through the redd can allow metabolic wastes to accumulate and kill eggs. Deposition of fine sediment can prevent newly hatched fish from emerging.
- Salmonids go through several life stage transformations while in their redds. The most important of these is the transformation from eggs into alevins, which have respiratory and movement capabilities and, hence, different vulnerability to being dewatered.

Redds are modeled using the following four daily actions. Scheduling of these actions is discussed in Section 8.

5.1. Survival

In *inSALMO*, eggs incubating in a redd are subject to five mortality sources: low and high temperatures, scouring by high flows, dewatering, and superimposition (having another redd laid on top of an existing one). Redd survival is modeled using redd “survival functions”, which determine, for each redd on each day, the probability of each egg surviving one particular kind of mortality. Then, a random draw is made on a binomial distribution to determine how many eggs survive each redd mortality source. A binomial distribution is a statistical model of the (integer) number of occurrences of some event within a specified number of trials, when the probability of occurrence per trial is known. In this case, the event is death of one egg, the number of trials is the number of eggs in the redd, and the probability of occurrence is one minus the survival function value. Hence, the binomial distribution returns a randomly drawn number of eggs that die, given the number of live eggs and the per-egg mortality probability. (The alternative approach of multiplying the mortality probability by the number of live eggs may appear simpler, but introduces a number of numerical difficulties when the number of live eggs is small.)

The separate redd mortality sources are executed sequentially: the eggs killed by one source are subtracted from the number alive before the next source is processed. The order in which redd survival functions are evaluated is defined in Section 8.3.

The kinds of mortality represented, and the survival function methods, were selected considering that the objectives of *inSALMO* focus on flow and temperature effects on salmon populations. Consequently, the methods are simple and focussed on temperature and flow effects. For example, there is no redd survival function related to spawning gravel quality. Spawning gravel quality has several effects on redd success (Kondolf 2000) but *inSALMO* is not designed address to represent gravel quality issues. [The spawning site selection criteria (Section 4.1.2) allow a fish to spawn in a cell that has little or no gravel; there is no redd mortality penalty for doing so. The exception is that if superimposition occurs in a cell with little spawning gravel (unlikely unless gravel is rare) then superimposition mortality is likely to be high.] For several of the redd mortality sources (especially, dewatering and superimposition), more detailed and mechanistic approaches are available in the literature and could be added to *inSALMO* in situations where these mortality sources are believed to be important.

5.1.1. Dewatering

Dewatering mortality occurs when flow decreases until a redd is no longer submerged; eggs can be killed by dessication or the buildup of waste products that are no longer flushed away. 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 *inSALMO* 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 is equal to the fish parameter *mortReddDewaterSurv*. 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, so *mortReddDewaterSurv* should not be too low.

5.1.2. Scouring and deposition

Scouring and deposition mortality results from high flows disturbing the gravel containing a redd. If eggs are scoured out of a redd, they likely to be washed downstream and are vulnerable to being eaten. Deposition of new gravel on top of a redd may make water flow through the redd inadequate to transport oxygen and waste materials, or may prevent newly hatched salmon from emerging. Deposition is especially likely to reduce survival if it includes fine sediment.

There are empirical methods for predicting the potential for scouring as a function of shear stress and substrate particle size at the local scale of a habitat cell, but geomorphologists now understand that scour and deposition at the scale of individual redds is a highly variable process best represented as stochastic. At least in gravel-bed streams, it is virtually impossible to predict where scour and deposition will occur at various flows (Haschenburger 1999, Wilcock et al. 1996). Consequently, *inSALMO* adopts an approach for predicting the probability of redd scouring or deposition from the empirical, reach-scale work of Haschenburger (1999). This approach was developed for gravel-bed channels and may not be appropriate for sites where spawning gravels occur mainly in pockets behind obstructions (where scouring is likely even less predictable). *inSALMO* should be considered substantially more uncertain for sites where populations are strongly limited by redd scouring, especially if spawning is limited to pocket gravels.

Haschenburger (1999) observed the spatial distribution and depth of scouring and deposition at a number of flow peaks in several study sites in gravel-bed rivers. The proportion of a stream reach that scoured or filled to a specified depth during a high-flow event was found to follow an exponential distribution, the parameter for which (*scourParam*) varies with site-average dimensionless (Shields) shear stress. Therefore, *inSALMO* assumes that the probability of a redd being destroyed is equal to the proportion of the stream reach scouring or filling to depths greater than the value of the fish parameter *mortReddScourDepth* (cm). Consequently, the probability of a redd *not* being destroyed (*scourSurvival*) is equal to the proportion of the stream scouring or filling to a depth less than the value of *mortReddScourDepth*. 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:

$$scourSurvival = 1 - e^{-scourParam \times mortReddScourDepth}.$$

(The value of *scourSurvival* is set to 1.0 if *scourParam*×*mortReddScourDepth* is greater than 100. This allows users to effectively turn scouring and deposition mortality off by using a very large value of *mortReddScourDepth*, e.g., 10,000 cm, without risk of the exponential function producing a variable underflow.)

The value of *scourParam* was modeled by Haschenburger empirically:

$$scourParam = 3.33 \times e^{-1.52 \times (shearStress/0.045)}.$$

where *shearStress* is the peak Shields stress (measured at a reach scale) occurring during the high-flow event. Shields stress is a dimensionless indicator of scour potential often used in modeling sediment transport, described in the sediment transport literature. Shields stress increases with flow, a relationship represented in *inSALMO* by the equation:

$$shearStress = habShearParamA \times flow^{habShearParamB}$$

where *habShearParamA* (s/m^3) and *habShearParamB* (unitless) are habitat reach parameters. These are habitat parameters because they are highly specific to each reach. Methods for estimating *habShearParamA* and *habShearParamB* are discussed in Section 16.7.2 of Railsback et al. (2009).

The fish parameter *mortReddScourDepth* 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 is almost certain to flush eggs out of the redd. Deposition of new material to this distance would double the egg pocket's depth, likely to severely reduce the survival and emergence of its eggs. DeVries (1997) reviews egg burial depths for stream salmonids. A value of 20 cm is reasonable for Chinook salmon.

Example scour survival parameters for a medium-gradient site with average gravel size of 3.5 cm (*habShearParamA* = 0.013, *habShearParamB* = 0.40, *mortReddScourDepth* = 20 cm) produce the relation between peak flow and probability of redd scouring illustrated in Figure 34.

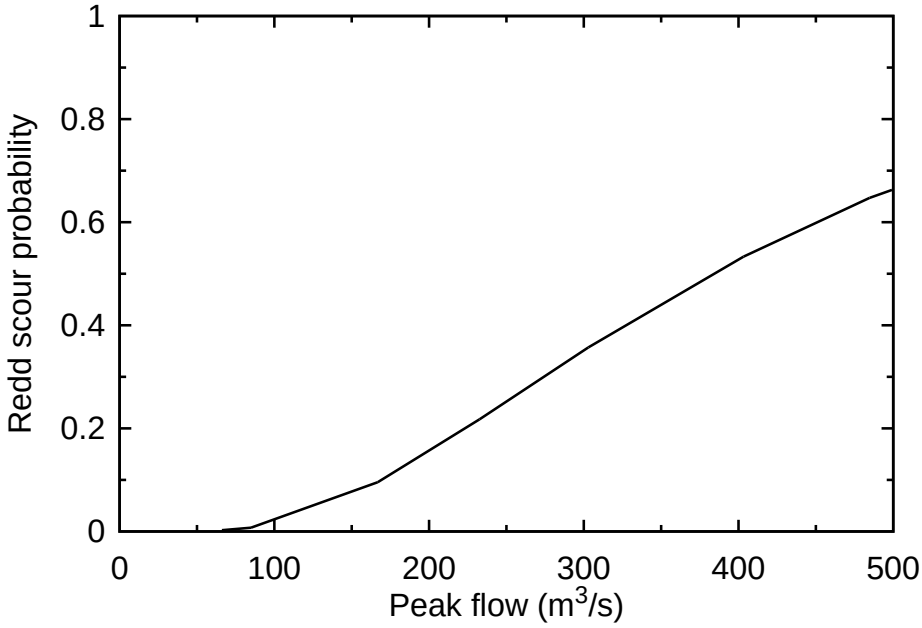


Figure 34. Example redd scour and fill survival function. The Y axis is the probability of a redd being destroyed during a peak flow event.

This model of scouring estimates the probability of a redd surviving scour in each high-flow event, not on a daily time step. The single survival probability is applied to all redds, assuming that if scouring occurs, then no eggs survive. [It is important to note that *inSALMO* calculates scouring survival from daily mean flows, whereas Haschenburger (1999) based her model on instantaneous peak flows. This approximation is made to avoid needing to input daily peak flows, but will cause scouring mortality to be underestimated when runoff is rapid.] The following steps are used for each redd, on each day.

- Determine whether the current day's flow in the redd's reach is greater than both the previous day's and the following day's flow. If so, then the following steps are conducted. If not, then the fraction of eggs surviving is 1.0 (no eggs are lost).

- Calculate the value of *scourSurvival*, using the above equations and the current day's flow for the redd's reach.

- Draw a uniform random number between zero and one. If the value of this random number is greater than the value of *scourSurvival*, then the fraction of eggs surviving is zero. Otherwise, the fraction of eggs surviving is 1.0.

To avoid the need for flow data for the date preceeding the start of a model run, redd scour is not executed on the first day of a run. However, redd scour can be executed on the last day, so flow input must extend at least one day past the last simulation date.

(Section 3.1.1 concerning flow input includes a potentially important note about using daily peak instead of mean flows to more accurately represent redd scour.)

5.1.3. Low temperature

Both low and high temperatures cause mortality in eggs, at temperatures much different than those causing mortality in fish. Mortality due to high and low temperatures are modeled separately. Logistic functions represent the available data well.

The daily fraction of eggs surviving low temperatures is modeled as an increasing logistic function of temperature. Parameter values appear to differ among species, with differences especially likely between species (or stocks) that spawn in the fall v. spring. In developing parameter values from published data on egg survival, it is important to remember that eggs incubate slowly at low temperatures, so even apparently high daily survival rates can result in low egg survival over the entire incubation period. Parameter values for fall Chinook salmon (Table 16; Figure 35) have been determined from data compiled by Myrick and Cech (2004).

Table 16. Parameter values for low temperature redd mortality.

Parameter	Definition	Species	Value
mortReddLoTT1	Temperature at which low temperature survival of eggs is 10 pct (°C)	Chinook	1.7
mortReddLoTT9	Temperature at which low temperature survival of eggs is 90 pct (°C)	Chinook	4.0

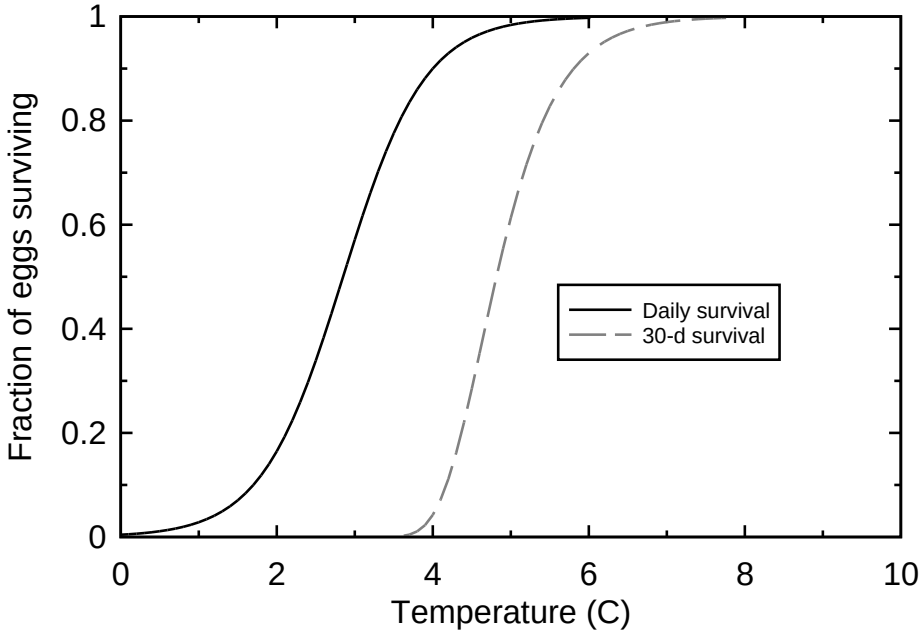


Figure 35. Low temperature redd survival function, for fall Chinook salmon parameter values.

5.1.4. High temperature

High temperatures can induce direct mortality in trout eggs, and also promote fungus and disease. The fraction of eggs surviving high temperatures is modeled as a decreasing logistic function of temperature (Figure 36). Parameter values for Chinook salmon were estimated to reproduce several patterns summarized from the literature by Myrick and Cech (2004): temperature effects become detectable around 12°, mortality is clear but not rapid by 14°, and mortality is strong and rapid by around 16.5°. The parameter values (Table 17) appear to indicate high survival at high temperatures, but in fact cause low survival if temperatures are elevated for long periods.

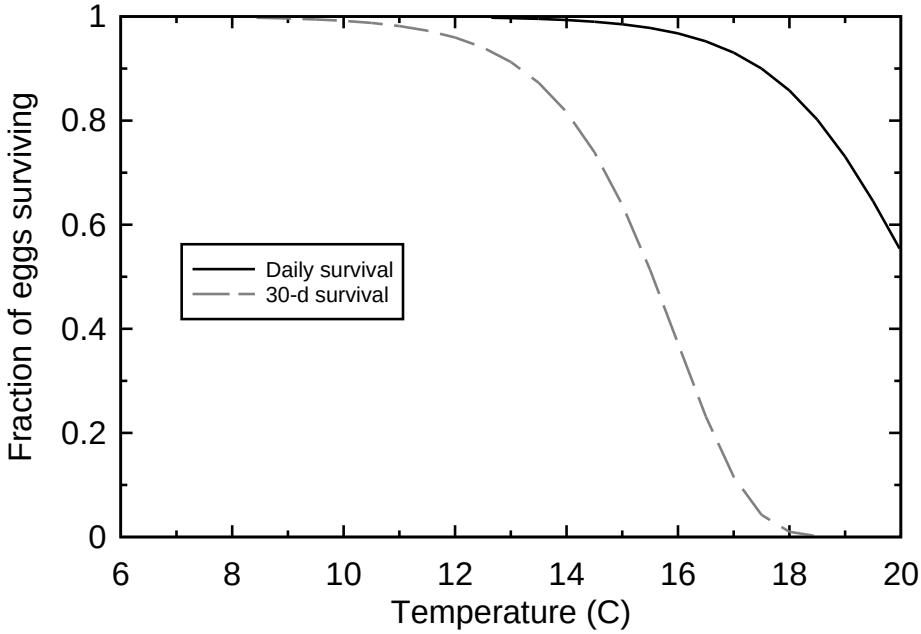


Figure 36. High temperature redd survival function, for Chinook salmon parameter values.

Table 17. Parameter values for high temperature redd mortality.

Parameter	Definition	Species	Value
mortReddHiTT1	Temperature at which high temperature survival of eggs is 10 pct (°C)	Chinook	23
mortReddHiTT9	Temperature at which high temperature survival is 90 pct (°C)	Chinook	17.5

5.1.5. Superimposition

Superimposition redd mortality can occur when a new redd is laid over an existing one; females digging new redds can disturb existing redds and cause egg mortality through mechanical damage or by displacing eggs from the redd environment. It is believed that superimposition typically causes mortality of many but not all eggs in a redd (Essington et al. 2000, Hendry et al. 2003). *inSALMO* assumes that superimposition is accidental with no bias for or against spawning over existing redds. The study by Essington et al. (1998) indicates that stream trout may indeed 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 that it might have, but

there is currently little known about what factors (e.g., sediment quality, spawner density) might encourage intentional superimposition.

Superimposition redd mortality is modeled as a function of the area disturbed in creating the new redd and the area of spawning gravel available. The following steps are executed by each redd, on each day it exists:

1. Determine if one or more new redds were created in the same cell on the current day. If not, then superimposition survival is 1.0 and the remaining steps are skipped.

2. In the event that *cellFracSpawn* is zero, there is no risk of superimposition and the remaining steps are skipped. This assumption is made because there is no gravel to be disturbed by another spawner.

3. If the female that created the redd is still alive, then the redd is assumed to be defended and superimposition survival is 1.0 and the remaining steps are skipped.

4. If one or more redds (of any species) were created in the same cell on the current day, the probability of them causing superimposition (*reddSuperImpRisk*, unitless) is equal to the total area of these new redds divided by the area of undefended spawning gravel in the redd.

$$reddSuperImpRisk = \frac{newReddsArea}{\left(cellArea \times cellFracSpawn\right) - defendedArea}$$

   The total area of new redds (*newReddsArea*) is the sum, over all redds created in the cell on the current day, of *reddSize*. The fish parameter *reddSize* (cm^2) is a characteristic area of the redd pit and tailspill, and is species-specific. The cell area defended by spawners (*defendedArea*) is the sum, over all live females who have already spawned (including those who spawned on the current day) of the fish parameter *fishSpawnDefenseArea*. This parameter represents the typical area (cm^2) that a female defends around her redd. If the denominator in the equation for *reddSuperImpRisk* is zero, or less than *newReddsArea* (so *reddSuperImpRisk* would be greater than one), then *reddSuperImpRisk* is set to 1.0.

5. A random number is drawn from a uniform distribution between zero and one; if it is less than *reddSuperImpRisk*, then superimposition mortality occurs.

6. If superimposition mortality occurs, then the fraction of eggs surviving is the value of another random number drawn from a uniform distribution between zero and one.

The parameter *reddSize* is defined as the area a spawner disturbs in creating a new redd. For Chinook salmon, a value of 56,000 cm^2 is supported by observations in the lower Columbia River basin (Burner 1951) and in Clear Creek, Sacramento River basin (Newton and Brown 2004). Burner (1951) noted that spawners tended to defend an area about four times the area of their redds, so a corresponding value of *fishSpawnDefenseArea* is 200,000 cm^2.

5.2. Development

To predict the timing of emergence, the developmental status of a redd's eggs is updated daily. Model redds accumulate the fractional development that occurs each day (*reddDailyDevel*), a function of temperature. This means the redd has a variable *fracDeveloped* that starts at zero when the redd is created and is increased each day by the value of daily value of *reddDailyDevel*. When *fracDeveloped* reaches 1.0, then the eggs are ready to emerge. The daily value of *reddDailyDevel* is determined using the equation recommended by Beacham and Murray (1990; their Model 4):

$$reddDailyDevel = \frac{1}{reddDevelParamA \times \left(temperature - reddDevelParamB\right)^{reddDevelParamC}}$$

Parameter values for chinook salmon from Beacham and Murray (1990; Table A.3 for fry emergence) are: *reddDevelParamA* = 33,000; *reddDevelParamB* = -2.04; *reddDevelParamC* = -7.58. The effect of temperature on development time for these parameter values is illustrated in Figure 37.

(The analysis of Beacham and Murray 1990 also indicates that eggs hatch into alevins when redd development reaches 0.6.)

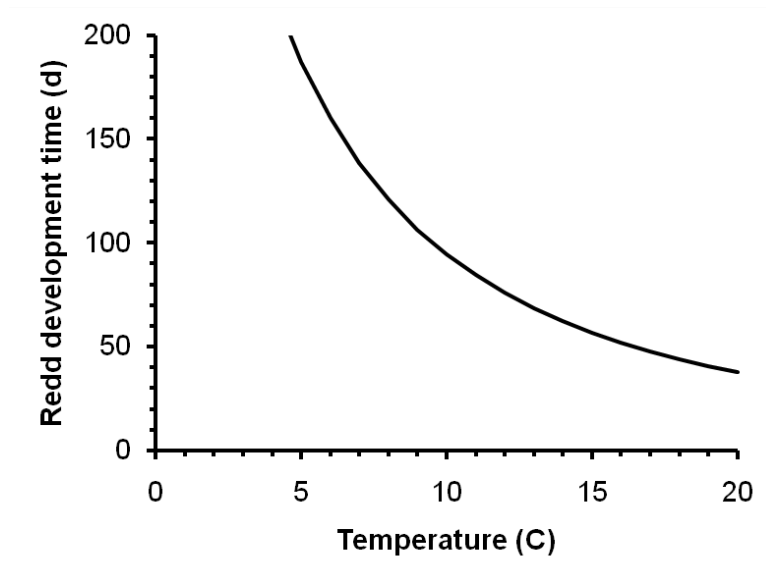


Figure 37. Redd development time (days from redd creation to emergence) vs. temperature for chinook salmon; the graph depicts the denominator of the equation for *reddDailyDevel*.

5.3. Emergence

"Emergence" is the conversion of each surviving egg into a new juvenile salmon object. When a redd's value of fracDeveloped equals or exceeds 1.0, its eggs are considered fully developed and ready to emerge as new fish. New fish emerge over several days. The following steps are used to determine how many fish emerge each day.

5.3.1. Emergence timing

Emergence begins on the day when *fracDeveloped* 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 move, competition would probably be overestimated. As a simple way to spread emergence over several days, *inSALMO* assumes that 10% of the redd's eggs 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 100 eggs on the day that development is complete, 10 new trout will be created on that day and 90 eggs will remain. On the next day (assuming no egg mortality occurs), 18 new fish will be created (20% of 90) and 72 eggs (90-18) remain in the redd. On the third day of emergence, 21 eggs (30% of 72, truncated to an integer) emerge. As emergence proceeds, the eggs remaining in a redd remain susceptible to egg mortality.

5.3.2. New fish attributes

For each new fish created from an egg that emerges, the model assigns these attributes.

- The fish is assigned its species from that of the redd.
- The fish is placed in the same habitat cell as its redd.
- Sex is assigned randomly, with even probability of being male or female.
- The length of each individual fish (*fishLength*, cm) is assigned from a uniform random distribution with minimum length equal to the fish parameter *reddNewLengthMin* (cm) and maximum length equal to the parameter *reddNewLengthMax* (cm).
- Weight (*fishWeight*, g) is calculated from length, using the length-weight relationship and parameters used in modeling growth (Section 4.3.1) and to create initial fish (Section 6.2). Fish are assumed to have a normal condition factor (*fishCondition* = 1.0) when they emerge:

$$fishWeight = fishWeightParamA \times fishLength^{fishWeightParamB}.$$

Variation among individuals in length at emergence is represented because habitat selection (and, consequently, growth and survival) is modeled using a length-based hierarchy (Section 4.2.1). Elliott (1994) found fish emerging from a redd to vary in size only slightly; but the variation gives larger fish an advantage in dominance that is likely to persist and grow over time because competition among newly emerged fish is often intense.

Example length parameter values for newly emerged Chinook salmon are provided in Table 18. These values were estimated from unpublished data provided by Michael Sparkman (California Department of Fish & Game, Arcata, CA) from four emergence traps deployed in the Redwood Creek drainage, Humboldt County, 1996-99.

Table 18. Parameter values for size of newly emerged fish.

Parameter	Definition	Chinook salmon value
reddNewLengthMin	Minimum of uniform distribution from which new fish lengths are drawn (cm)	3.5
reddNewLengthMax	Maximum of uniform distribution from which new fish lengths are drawn (cm)	4.1

5.3.3. Superindividual subadults

When juvenile salmon are created upon emergence from redds, they optionally can be modeled as "superindividuals" to reduce computations and significantly reduce the time that *inSALMO* takes to execute. Superindividuals are model objects that behave exactly as a normal fish but represent multiple individuals. Each fish in *inSALMO* has an instance variable *nREP*, an integer for how many fish are represented by it. Adults are initialized with *nREP* set to 1 so they always represent just one fish.

When juveniles are created upon emergence from their redd, their value of nREP is set equal to a model variable *juvenileSuperindividualRatio*, which is given a value in the Model.Setup file. If, for example, *juvenileSuperindividualRatio* is set to 10 in Model.Setup, then the number of new fish created from a redd is about 1/10th the number of eggs that hatch, and each such fish represents 10 real fish. The following methods are used.

- On days when its is fully developed, a redd calculates how many eggs emerge as new fish (Section 5.3.1).

- If this number of emerging eggs is less than the value of *juvenileSuperindividualRatio*, then one new fish object is created with its value of *nREP* set to the value of *juvenileSuperindividualRatio*. The redd is then assumed to be empty.

- If instead the number of emerging eggs is greater than the value of *juvenileSuperindividualRatio*, then the number of new fish objects created is calculated by dividing the number of emerging eggs by *juvenileSuperindividualRatio* and rounding the result to an integer. This number of new fish objects are created and their value of *nREP* set to the value of *juvenileSuperindividualRatio*. The number of eggs remaining in the redd is set by subtracting the number of emerging eggs (not the number of new fish objects times *juvenileSuperindividualRatio*) from it.

- During its habitat selection method, a fish compares the available drift and search food in potential destination cells to the food it would consume in the current day. In this method the food the fish would consume is multiplied by its value of *nREP* to account for the multiple fish it represents.

- When a fish occupies a cell and feeds there, the food and velocity shelter it consumes is multiplied by *nREP*.

- The file output for number of live and dead fish are modified so instead of reporting the count of fish objects they report the sum of *nREP* over the fish. These outputs therefore report the number of represented fish, not the number of fish objects. (However, the graphical outputs, including the animation raster and the graphs, report the number of fish objects, not multiplied by *nREP*.) File output for biomass are likewise adjusted for *nREP* so they report the total mass of fish represented (biomass output sums the product of individual weight and *nREP* over all fish objects).

It is important to understand the effects of superindividuals on model results: this technique can introduce biases that are difficult to predict. To understand how superindividuals affect results, an experiment ran inSALMO for *juvenileSuperindividualRatio* values of 1, 5, 10, 20, 50, and 100; with 50 replicate simulations of each value. The model was run for one year using two study sites. Results indicate that values of *juvenileSuperindividualRatio* above 20 can have relatively strong effects on abundance, especially at the end of the season when juvenile numbers become low (Figure 38). Likewise, high values of *juvenileSuperindividualRatio* can have strong effects on juvenile lengths during periods when abundance is low (Figure 39).

An experiment with a simple test program, in which superindividuals do nothing but survive daily with probability of 0.99, produced an increase in mean abundance with increasing superindividual ratio similar to that in Figure 38. This experiment indicates that the effect of superindividual ratio is largely an artifact of the survival mathematics. The effect of *juvenileSuperindividualRatio* on length likely results from changes in abundance, which affect competition for feeding sites.

Length of the few fish that migrate out latest can be a very important outcome of *inSALMO*, so artifacts of *juvenileSuperindividualRatio* should be carefully avoided. Values above 20 appear risky, while values of 10 and lower had little apparent effect in this experiment.

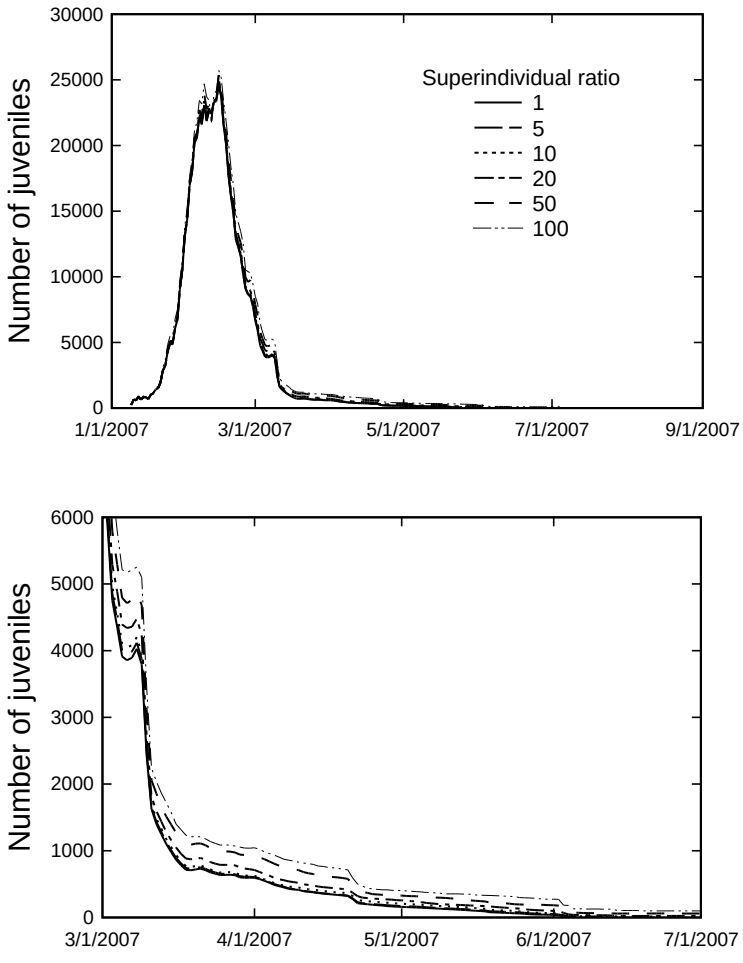


Figure 38. Effect of *juvenileSuperindividualRatio* on juvenile abundance. Results are means of 50 replicates. Panel b is simply a zoom in on March 1 to July 1 results.

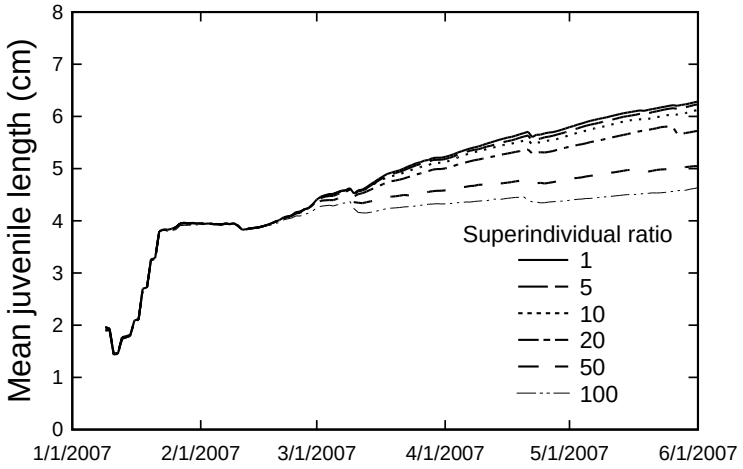


Figure 39. Effect of *juvenileSuperindividualRatio* on juvenile mean length. Results are means of 50 replicates.

5.4. Empty Redds

As described in the previous sections, the number of eggs remaining in redds is reduced when eggs die or fish emerge. When the number of remaining eggs in a redd reaches zero, the redd is dropped from the model.

6. Initialization

This section describes the methods used to initialize the habitat and fish populations when each new model run is started. Because *inSALMO* 1.0 is designed to simulate one or more seaasons of spawning through outmigration, the start of a simulation is typically just before adults arrive via migration from the ocean, with no salmon present. Although this section mentions some of the input types and files, complete documentation of file and input types is provided in a separate software document.

6.1. Habitat Initialization

A model run starts by reading in the habitat characteristics that do not change during the simulation. These characteristics are the number of reaches and how they are linked, the location and dimensions of cells in each reach, the values of cell variables that do not change with time, and the lookup tables used to calculate daily depth and velocity in each cell (Section 3.2). Finally, variables that depend on time-series input (reach temperature, flow, turbidity; cell depth and velocity) are initialized with the input data for the first simulation date.

6.2. Fish Initialization

At the start of a simulation, *inSALMO* creates the adult spawners that then “arrive”—are added to the model at their designated reach—over time as the simulation proceeds. These adults are the only salmon initialized.

6.2.1. Spawner arrival dates

The number of adults and their initial characteristics are specified as model input, via a population initialization file. These initial characteristics include the date at which each adult "arrives" and enters the simulation. The method for distributing spawner arrival dates is taken from *inSALMO* 0.5, which assumes a truncated normal distribution. The parameters for the distribution of arrival dates are provided as part of the population initialization input for each species, reach, and year. The initialization input variables *arrivalStartDate* and *arrivalEndDate* (in MM/DD/YYYY format) specify the first and last dates of arrival. The peak date of arrival is by definition halfway between these first and last dates. The variable *arrivalRatio* specifies the ratio of the distribution's peak height (at the middle of the arrival period) to the distribution's height on the first and last arrival days (Figure 40). The algorithm to assign arrival dates to spawners is:

- The first and last arrival dates are set to *arrivalStartDate* and *arrivalEndDate*.
- The mean (peak) arrival date is set equal to the mean of *arrivalStartDate* and *arrivalEndDate*.
- The standard deviation in arrival dates is set to the length of the arrival period (difference between last and first days) divided by $2[-2\ln(\textit{arrivalRatio})]^{0.5}$.
- For each spawner initialized, a random value is drawn from a normal distribution with the above mean and standard deviation, then rounded to an integer date. If this date is before the first arrival date or after the last, it is discarded and another value drawn.

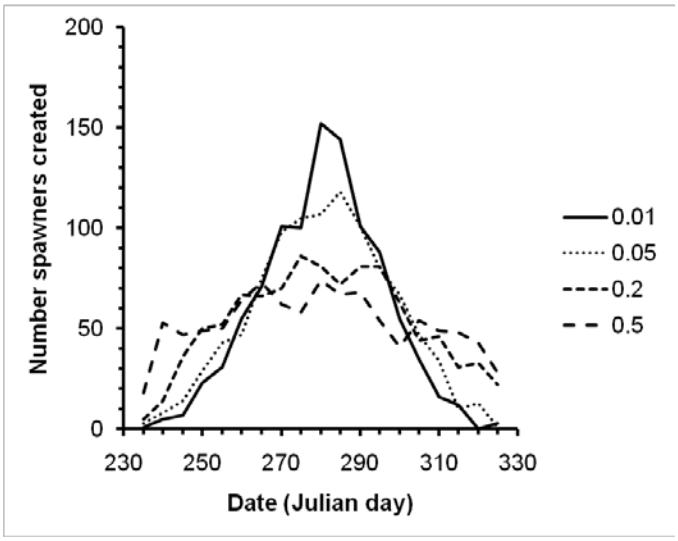


Figure 40. Example distributions of spawner arrivals (number of arrivals in 5-day periods, out of 1000 total adults) for four values of *fishArrivalRatio* (0.01–0.5), with *fishArrivalStartDate* and *fishArrivalEndDate* set to 8/26/2005 (Julian day 238) and 11/25/2005 (Julian day 329).

6.2.2. Initialization of spawner numbers, locations, and variables

The number and sex distribution of initial spawners is specified by input. Separate input values for each species, year, and reach specify the number of spawners to create and what fraction of

them are female. If, for example, 200 spawners are to be created for a reach and the fraction female is 0.6, then exactly 120 females and 80 males are created.

Spawner lengths are drawn randomly from a normal distribution with mean and standard deviation specified as input (also for each species, year, and reach). Because length can differ between males and females and only the female's length matters to the model (because it determines fecundity), these spawner length parameters should represent females only. The weight of each fish is calculated from its length using parameters *fishWeightParamA* and *fishWeightParamB* and the method used for new juveniles (Section 5.3.2).

In this version of *inSALMO*, spawner ages do not need to be simulated. Instead of removing the age variable (it may be needed in subsequent versions of the model) or specifying it in input (making the input more complex), the age of every spawner is simply set to 5 years.

When adults are added to the model on their arrival date, they are placed in a cell chosen randomly from among those having depth greater than half the spawner's length.

Fish have two variables controlling when and if they spawn. The variable *isSpawner* is initialized to YES, indicating that all adults are eligible to spawn during the current spawning season. The variable *spawnedThisSeason* indicates whether the fish has already spawned during the current spawning season (Section 4.1.1) and is initialized to NO.

6.3. Redd Initialization

There is no capability in *inSALMO* to initialize redds at the start of a simulation. Redds can only be created by spawning fish.

7. Random Number Generation

Several processes in *inSALMO* (e.g., fish initialization; fish survival) are modeled stochastically, using pseudo-random numbers to determine process outcomes. How pseudo-random numbers are generated is an important issue for any stochastic simulation model, as poor quality or mis-used random number generators can bias simulation results.

All pseudo-random numbers in *inSALMO* are generated by the MT19937 "Mersenne Twister" algorithm, the default generator in the Swarm software platform used to implement *inSALMO*. (See SDG 2000 for additional information and references.)

One random number generator is used for all stochastic processes in *inSALMO*. This generator is initialized with a random number seed, *randGenSeed*, provided by the user as a model parameter. If two model runs use the same value of *randGenSeed* <u>and</u> exactly the same input and parameters, the two runs will produce exactly the same results. However, any change to input (parameter values, input data, simulation dates, etc.) is very likely to alter the number of times the random number generator is called and, therefore, the outcome of all stochastic processes. Replicate simulations are produced by altering only the value of *randGenSeed*.

8. Scheduling

The order in which events occur can strongly affect the outcome of individual-based models. This section defines the schedule by which the events in *inSALMO* are executed. The schedule

consists of an ordered list of actions, each executed once per simulation day. An action is defined by a list of objects, the methods those objects execute, and rules for the order in which the objects are processed. There are four main action groups (groups of related actions over the same list of objects): habitat, fish, redd, and observer. The full schedule is displayed at the end of this section.

8.1. Habitat Update Actions

Habitat updates are scheduled first because subsequent fish and redd actions depend on the day's habitat conditions. For each reach, time-series input data (flow, temperature, turbidity) are obtained for the current simulation date. The new flow is used to update the depth and velocity of all cells in each reach. The daily food production is calculated for each cell, and the amount consumed by fish is reset to zero.

8.2. Fish Actions

Fish actions are scheduled before redd actions because one fish action (spawning) can cause redd mortality via superimposition. This order means that new fish emerging from a redd do not execute their first fish actions until the day after their emergence. Scheduling fish spawning before redd actions also means that redds undergo all redd actions on the day they are created.

The four fish actions in the model are conducted in the following order: spawning, habitat selection, growth, and survival. Actions are carried out one fish at a time, in descending order of fish length. Each of these four actions is conducted for all fish before the next action is executed.

Spawning is the first fish action because spawning can be assumed the primary activity of a fish on the day it spawns. Spawning also affects habitat selection in two ways. First, female spawners move to a cell with spawning habitat on the day they create a redd. Second, when fish spawn their weight and condition are substantially reduced, which affects their choice of habitat (giving higher preference to habitat providing high growth).

Habitat selection is the second fish action each day because it is the way that fish adapt to the day's new habitat conditions; habitat selection strongly affects both growth and survival. Note that habitat selection is affected by fish size and condition (which affect survival probabilities and reproductive status). Habitat selection is based the fish's size *before* the current day's growth, because a fish's growth depends on its habitat choice.

Growth is scheduled before survival because changes in a fish's length or condition factor affect its probability of survival.

The last fish action is survival. Survival has its own sub-schedule because it includes evaluation of several different mortality sources. The number of fish killed by each mortality source can be affected by the order in which survival probabilities for each source are evaluated. Placing a mortality source earlier in the survival sub-schedule makes it slightly more likely to cause mortality (a mortality source cannot kill a given fish on a given day if a preceding mortality source kills the fish first). Therefore, widespread, less random mortality sources (e.g., high temperatures, high velocities) are scheduled first; survival probabilities for these sources tend to be negligible (very close to 1.0) under most conditions and low (causing high mortality) when an unusual event occurs.

8.3. Redd Actions

Redd actions occur last each day because redds do not affect either habitat cells or fish (with the exception of creating new fish, as discussed above). There are three redd actions: survival, development, and emergence. These actions are applied to the existing redds in the order in which the redds were created, but this order has no effect on redds or newly emerged trout.

Redd survival is the first redd action to be executed. Survival is scheduled before emergence so that eggs are subject to mortality on the day they emerge; otherwise, emerging fish would risk neither redd mortality nor fish mortality for one lucky day. Redd survival includes five separate egg mortality sources which follow their own sub-schedule. The redd mortality sources are scheduled from least random (extreme temperatures) to most random (superimposition).

Development is the second redd action, and emergence is third. Because development preceeds emergence, new fish begin to emerge from a redd on the same day the redd's egg development is complete.

8.4. Observer Actions

Observer actions collect and record data on the digital world inside *inSALMO*. Because the output produced by observer actions is the only information that users have about the complex events going on inside the model, fully understanding model results requires knowing how observations are scheduled with respect to other model actions.

Observer actions are the last of the daily model actions. Therefore, the model's graphical and file outputs represent the state of the model after all the habitat, fish, and redd actions have been completed for a day. This scheduling means, for example, that the size and condition of a fish observed from the graphical user interface reflects the fish's state after it has completed its daily feeding and growth, not its state when it made its habitat selection decision or when it considered spawning.

8.5. Complete Schedule

The complete schedule is displayed in Figure 41. This figure displays the four main action groups and the actions within each group, in the order they are executed on each daily time step.

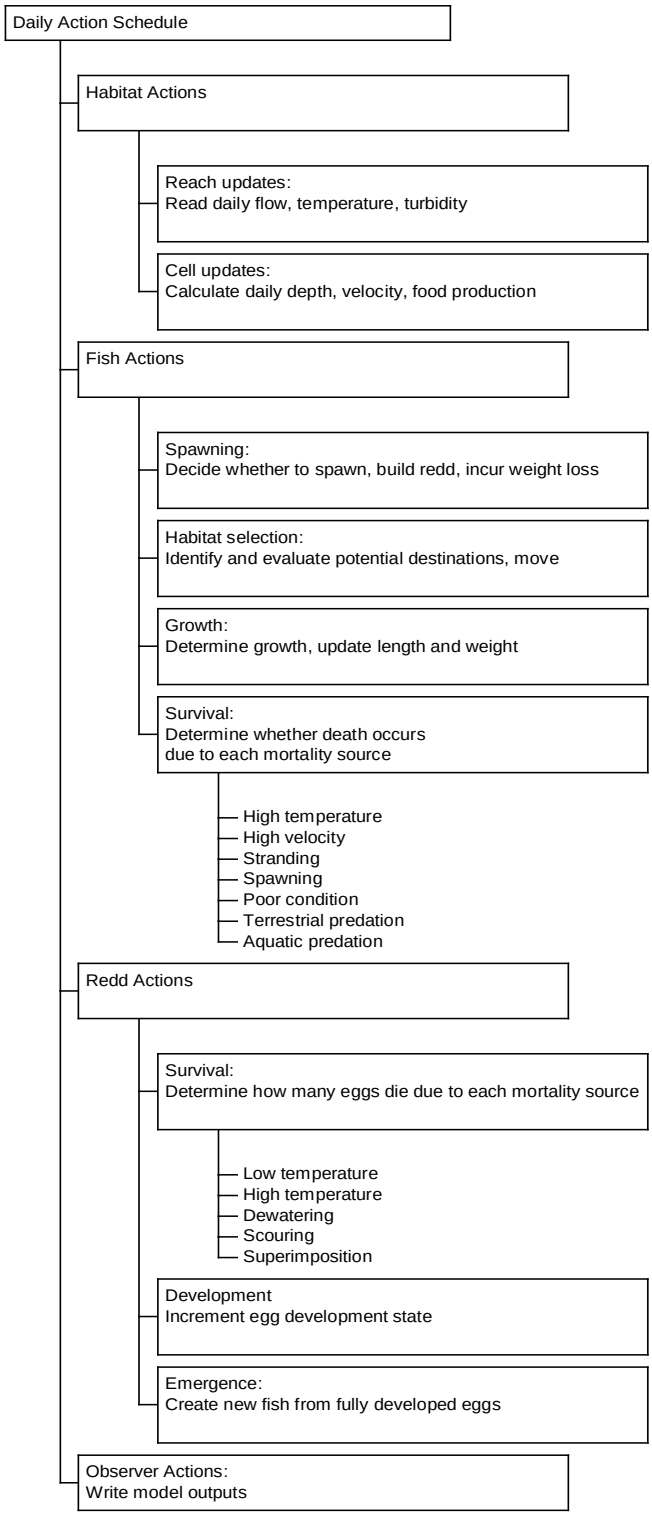


Figure 41. Complete schedule of daily actions.

9. Literature Cited

Alexander, G. R. 1979. Predators of fish in coldwater streams. Pages 153-170 *in* H. Clepper, editor. Predator-prey systems in fisheries management. Sport Fishing Institutes, Washington, D. C.

Alsop, D. H., and C. M. Wood. 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.

Baltz, D. M.; Moyle, P. B. 1984. Segregation by species and size classes of rainbow trout, *Salmo gairdneri*, and Sacramento sucker, *Catostomus occidentalis*, in three California streams. Environmental Biology of Fishes 10: 101-110.

Baltz, D. M.; Vondracek, B.; Brown, L. R.; Moyle, P. B. 1987. Influence of temperature on microhabitat choice by fishes in a California stream. Transactions of the American Fisheries Society 116: 12-20.

Barrett, J. C., G. D. Grossman, and J. Rosenfeld. 1992. Turbidity-induced changes in reactive distance of rainbow trout. Transactions of the American Fisheries Society **121**:437-443.

Bartell, S. M., J. M. Breck, R. H. Gardner, and A. L. Brenkert. 1986. Individual parameter perturbation and error analysis of fish bioenergetics models. Canadian Journal of Fisheries and Aquatic Sciences **43**:160-168.

Beacham, T. D., and C. B. Murray. 1990. Temperature, egg size and development of embryos and alevins of five species of Pacific salmon: a comparative analysis. Transactions of the American Fisheries Society **119**:927-945.

Behnke, R. J. 1992. Native trout of western North America. Bethesda MD: American Fisheries Society.

Booker, D. J., M. J. Dunbar, and A. Ibbotson. 2004. Predicting juvenile salmonid drift-feeding habitat quality using a three-dimensional hydraulic-bioenergetic model. Ecological Modelling **177**:157–177.

Braaten, P. J., P. D. Dey, and T. C. Annear. 1997. Development and evaluation of bioenergetic-based habitat suitability criteria for trout. Regulated Rivers: Research and Management **13**:345-356.

Bradford, M. J., and P. S. Higgins. 2001. Habitat-, season-, and size-specific variation in diel activity patterns of juvenile chinook salmon (*Oncorhynchus tshawytscha*) and steelhead trout (*Oncorhynchus mykiss*). Canadian Journal of Fisheries and Aquatic Sciences **58**:365-374.

Brandt, S. B., and K. J. Hartman. 1993. Innovative approaches with bioenergetics models: Future applications to fish ecology and management. Transactions of the American Fisheries Society **122**:731-735.

Brown, L. C.; Barnwell, T. O. J. 1987. The enhanced stream water quality models QUAL2E and QUAL2E-UNCAS: Documentation and user manual. EPA/600/3-87/07, Environmental Research Laboratory, U.S. Environmental Protection Agency, Athens GA.

Burner, C.J. 1951. Characteristics of spawning nests of Columbia River salmon. Fishery Bulletin 61, **52**:97-110.

Butler, P. J., N. Day, and K. Namba. 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.

Clark, C. W., and M. Mangel. 2000. Dynamic state variable models in ecology. Oxford University Press, New York.

Cunjak, R. A., T. D. Prowse, and D. L. Parrish. 1998. Atlantic salmon (*Salmo salar*) in winter: "the season of parr discontent"? Canadian Journal of Fisheries and Aquatic Sciences **55(Suppl. 1)**:161-180.

DeRobertis, A., C. H. Ryer, A. Veloza, and R. D. Brodeur. 2003. Differential effects of turbidity on prey consumption of piscivorous and planktivorous fish. Canadian Journal of Fisheries and Aquatic Sciences **60**:1517-1526.

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.

Dickerson, B. R., and G. L. Vinyard. 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.

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. Oxford University Press, New York.

Elliott, J. M., and M. A. Hurley. 2000. Daily energy intake and growth of piscivorous brown trout, *Salmo trutta*. Freshwater Biology **44**:237-245.

Essington, T. E., P. W. Sorensen, and D. G. Paron. 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.

Essington, T. E., T. P. Quinn, and V. E. Ewert. 2000. Intra- and inter-specific competition and the reproductive success of sympatric Pacific salmon. Canadian Journal of Fisheries and Aquatic Sciences **57**:205-213.

Fausch, K. D. 1984. Profitable stream positions for salmonids: relating specific growth rate to net energy gain. Canadian Journal of Zoology **62**:441-451.

Fraser, N., and N. B. Metcalfe. 1997. The costs of becoming nocturnal: feeding efficiency in relation to light intensity in juvenile Atlantic salmon. Functional Ecology **11**:385-391.

From, J., and G. Rasmussen. 1984. A growth model, gastric evacuation, and body composition in rainbow trout, *Salmo gairdneri* Richardson, 1836. Dana **3**:61-139.

Gard, M. 1997. Techniques for adjusting spawning depth habitat utilization curves for availability. Rivers **6**:94-102.

Gowan, C., and K. D. Fausch. 1996. Mobile brook trout in two high-elevation Colorado streams: re-evaluating the concept of restricted movement. Canadian Journal of Fisheries and Aquatic Sciences **53**:1370-1381.

Gowan, C., and K. D. Fausch. 2002. Why do foraging stream salmonids move during summer? Environmental Biology of Fishes **64**:139-153.

Gregory, R. S., and C. D. Levings. 1999. Turbidity reduces predation on migrating juvenile pacific salmon. Transactions of the American Fisheries Society **127**:275-285.

Griffiths, J. S., and D. F. Alderdice. 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., and S. F. Railsback. 2005. Individual-based modeling and ecology. Princeton University Press, Princeton, New Jersey.

Groot, C., and L. Margolis, editors. 1991. Pacific salmon life histories. UBC Press, Vancouver, Canada.

Grossman, G. D., P. A. Rincon, M. D. Farr, and R. E. J. Ratajczak. 2002. A new optimal foraging model predicts habitat use by drift-feeding stream minnows. Ecology of Freshwater Fish **11**:2-10.

Hanson, P., T. Johnson, J. Kitchell, and D. E. Schindler. 1997. Fish Bioenergetics 3.0. University of Wisconsin Sea Grant Institute, Madison, Wisconsin.

Harvey, B. C., and A. J. Stewart. 1991. Fish size and habitat depth relationships in headwater streams. Oecologia **87**:336-342.

Harvey, B. C., and C. D. Marti. 1993. The impact of dipper, *Cinclus mexicanus*, predation on stream benthos. Oikos **68**:431-436.

Harvey, B. C., R. J. Nakamoto, and J. L. White. 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**:2161-2166.

Haschenburger, J. K. 1999. A probability model of scour and fill depths in gravel-bed channels. Water Resources Research **35**:2857-2869.

Hawkins, D. K., and T. P. Quinn. 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., J. D. Stark, and K. A. Shearer. 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*). Pages 564 *in* C. Groot and L. Margolis, editors. Pacific Salmon Life Histories. UBC Press, Vancouver.

Healey, M. C., and W. R. Heard. 1984. Inter- and intra-population variation in the fecundity of chinook salmon (*Onchorynchus tshawytscha*) and its relevance to life history theory. Canadian Journal of Fisheries and Aquatic Sciences **41**:476-483.

Hendry, A. P., Y. E. Morbey, O. K. Berg, and J. K. Wenburg. 2003. Adaptive variation in senescence: reproductive lifespan in a wild salmon population. Proceedings of the Royal Society of London B **271**:259-266.

Hill, J., and G. D. Grossman. 1993. An energetic model of microhabitat use for rainbow trout and rosyside dace. Ecology **74**:685-698.

Hodgens, L. S., S. C. Blumenshine, and J. C. Bednarz. 2004. Great blue heron predation on stocked rainbow trout in an Arkansas tailwater fishery. North American Journal of Fisheries Management **24**:63-75.

Hokanson, K. E. F., C. F. Kleiner, and T. W. Thorslund. 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.

Houston, A. I., and J. M. McNamara. 1999. Models of adaptive behavior: an approach based on state. Cambridge University Press, Cambridge.

Hughes, N. F. 1992a. 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. 1992b. Ranking of feeding positions by drift-feeding arctic grayling (*Thymallus arcticus*) in dominance hierarchies. Canadian Journal of Fisheries and Aquatic Sciences **49**:1994-1998.

Hughes, N. F., and L. M. Dill. 1990. Position choice by drift-feeding salmonids: Model and test for arctic grayling (*Thymallus arcticus*) in subarctic mountain streams, interior Alaska. Canadian Journal of Fisheries and Aquatic Sciences **47**:2039-2048.

Hughes, N. F., J. W. Hayes, K. A. Shearer, and R. B. Young. 2003. Testing a model of drift-feeding using 3-dimensional videography of wild brown trout in a New Zealand river. Canadian Journal of Fisheries and Aquatic Sciences **60**:1462-1476.

Hurlbert, S. H. 1984. Pseudoreplication and the design of ecological field experiments. Ecological Monographs **54**:187-211.

Jones, M. W., and J. A. Hutchings. 2002. Individual variation in Atlantic salmon fertilization success: implications for effective population size. Ecological Applications **12**:184-193.

June, J. A. 1981. Life history and habitat utilization of cutthroat trout (*Salmo clarki*) in a headwater stream on the Olympic Peninsula, Washington. MS thesis, College of Fisheries, University of Washington, Seattle.

Keeley, E. R., and J. W. A. Grant. 2001. Prey size of salmonid fishes in streams, lakes, and oceans. Canadian Journal of Fisheries and Aquatic Sciences **58**:1122-1132.

Knapp, R.A., and H.K. Preisler. 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**: 1576-1584.

Kondolf, G. M. 2000. Some suggested guidelines for geomorphic aspects of anadromous salmonid habitat restoration proposals. Restoration Ecology **8**:48-56.

Lam, T. J. 1988. Environmental influences on gonadal activity in fish. pp. 65-116 in W. S. Hoar, D. J. Randall, and E. M. Donaldson, eds. Fish Physiology, Volume IX, Reproduction, Part B: Behavior and Fertility Control. Academic Press, New York.

MacNutt, M. J., S. G. Hinch, A. P. Farrell, and S. Topp. 2004. The effect of temperature and acclimation period on repeat swimming performance in cutthroat trout. Journal of Fish Biology **65**:342-353.

Mangel, M., and C. W. Clark. 1986. Toward a unified foraging theory. Ecology **67**:1127-1138.

Mesa, M. G., and C. D. Magie. 2006. Evaluation of energy expenditure in adult spring Chinook salmon migrating upstream in the Columbia River Basin: an assessment based on sequential proximate analysis. River Research and Applications 22:1085-1095.

Metcalfe, N. B., N. H. C. Fraser, and M. D. Burns. 1999. Food availability and the nocturnal vs. diurnal foraging trade-off in juvenile salmon. Journal of Animal Ecology **68**:371-381.

Meyer, K. A., D. J. Schill, F. S. Elle, and J. A. J. Lamansky. 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.

Morbey, Y. E., and R. C. Ydenberg. 2003. Timing games in the reproductive phenology of female Pacific salmon (*Oncorhynchus* spp.). American Naturalist **161**:284-298.

Morin, A., and P. Dumont. 1994. A simple model to estimate growth rate of lotic insect larvae and its value for estimating population and community production. Journal of the North American Benthological Society **13**:357-367.

Moyle, P. B., and D. M. Baltz. 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. PhD. University of California, Davis, Davis, CA.

Myrick, C. A., and J. J. Cech, Jr. 2000. Temperature influences on California rainbow trout physiological performance. Fish Physiology and Biochemistry **22**:245-254.

Myrick, C. A., and J. J. Cech, Jr. 2003. The physiological performance of golden trout at water temperatures of 10 - 19°C. California Fish and Game **89**:20-29.

Myrick, C. A., and J. J. Cech, 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.

Nelson, R.L., W.S. Platts, and O. Casey. 1987. Evidence for variability in spawning behavior of interior cutthroat trout in response to environmental uncertainty. Great Basin Naturalist **47**: 480-487.

Newton, J. M. and M. R. Brown. 2004. Adult Spring Chinook Salmon Monitoring in Clear Creek, California, 1999-2002. USFWS Report. US Fish and Wildlife Service, Red Bluff, CA.

Nielsen, J. L. 1992. Microhabitat-specific foraging behavior, diet, and growth of juvenile coho salmon. Transactions of the American Fisheries Society **121**:617-634.

Nislow, K., C. Folt, and M. Seandel. 1998. Food and foraging behavior in relation to microhabitat use and survival of age-0 Atlantic salmon. Canadian Journal of Fisheries and Aquatic Sciences **55**:116-127.

Pacific Gas and Electric Company (PG&E). 1994. Evaluation of factors causing variability in habitat suitability criteria for Sierra Nevada trout. Report 009.4-94.5. Pacific Gas and Electric Company, Department of Research and Development, San Ramon, CA.

Petrusso, P. A., and D. B. Hayes. 2001. Condition of juvenile Chinook salmon in the upper Sacramento River, California. California Fish and Game **87**:19-37.

Power, M. E. 1987. Predator avoidance by grazing fishes in temperate and tropical streams: importance of stream depth and prey size. Pages 333-352 *in* W. C. Kerfoot and A. Sih,

editors. Predation, Direct and Indirect Impacts on Aquatic Communities. University Press of New England, Hanover.

Preall, R. J., and N. H. Ringler. 1989. Comparison of actual and potential growth rates of brown trout (*Salmo Trutta*) in natural streams based on bioenergetic models. Canadian Journal of Fisheries and Aquatic Sciences **46**:1067-1076.

Quinn, T. P., and G. B. Buck. 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., and K. A. Rose. 1999. Bioenergetics modeling of stream trout growth: temperature and food consumption effects. Transactions of the American Fisheries Society **128**:241-256.

Railsback, S. F., R. H. Lamberson, B. C. Harvey, and W. E. Duffy. 1999. Movement rules for spatially explicit individual-based models of stream fish. Ecological Modelling **123**:73-89.

Railsback, S., and B. Harvey. 2001. Individual-based model formulation for cutthroat trout, Little Jones Creek, California. General Technical Report PSW-GTR-182, Pacific Southwest Research Station, Forest Service, U. S. Department of Agriculture, Albany, CA.

Railsback, S. F., and B. C. Harvey. 2002. Analysis of habitat selection rules using an individual-based model. Ecology **83**:1817-1830.

Railsback, S. F., B. C. Harvey, R. H. Lamberson, D. E. Lee, N. J. Claasen, and S. Yoshihara. 2002. Population-level analysis and validation of an individual-based cutthroat trout model. Natural Resource Modeling **15**:83-110.

Railsback, S. F., B. C. Harvey, J. W. Hayse, and K. E. LaGory. 2005. Tests of theory for diel variation in salmonid feeding activity and habitat use. Ecology **86**:947-959.

Railsback, S. F., B. C. Harvey, S. K. Jackson, and R. H. Lamberson. 2009. inSALMO: the individual-based stream trout research and environmental assessment model. General Technical Report PSW-GTR-218, USDA Forest Service, Pacific Southwest Research Station, Albany, California. Available at: http://www.fs.fed.us/psw/publications/documents/psw_gtr218/psw_gtr218.pdf

Railsback, S. F., B. C. Harvey, and J. L. White. 2011. inSALMO Version 1.0: Model Improvements and Demonstration application to Chinook salmon spawning, incubation, and rearing in Clear Creek, California. Report prepared by Lang, Railsback and Associates, Arcata, CA for US Bureau of Reclamation, Mid-Pacific Regional Office, Sacramento CA.

Rand, P. S., D. J. Stewart, P. W. Seelbach, M. L. Jones, and L. R. Wedge. 1993. Modeling steelhead population energetics in lakes Michigan and Ontario. Transactions of the American Fisheries Society **122**:977-1001.

Reiser, D. W., and R. G. White. 1983. Effects of complete redd dewatering on salmonid egg-hatching success and development of juveniles. Transactions of the American Fisheries Society **112**:532-540.

Schmidt, D., and W. J. O'Brien. 1982. Planktivorous feeding ecology of Arctic grayling (*Thymallus arcticus*). Canadian Journal of Fisheries and Aquatic Sciences **39**:475-482.

Schneider, M. J., and T. J. Connors. 1982. Effects of elevated water temperature on the critical swimming speeds of yearling rainbow trout, *Salmo gairdneri*. Journal of Thermal Biology **7**:227-229.

SDG. 2000. Document set for Swarm 2.1. Swarm Development Group, Santa Fe, New Mexico.

Simpkins, D. G., W. A. Hubert, C. Martinez Del Rio, and D. C. Rule. 2003a. 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.

Simpkins, D. G., W. A. Hubert, C. Martinez Del Rio, and D. C. Rule. 2003b. 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.

Sweka, J. A., and K. J. Hartman. 2001. Influence of turbidity on brook trout reactive distance and foraging success. Transactions of the American Fisheries Society **130**:138-146.

Taylor, S. E., S. Egginton, and E. W. Taylor. 1996. Seasonal temperature acclimatisation of rainbow trout: cardiovascular and morphometric influences on maximal sustainable exercise level. Journal of Experimental Biology **199**:835-845.

Thut, R. N. 1970. Feeding habits of the dipper in southwestern Washington. Condor **72**: 234-235.

US Fish and Wildlife Service (USFWS). 2011. Flow-habitat relationships for fall-run Chinook salmon and steelhead/rainbow trout spawning in Clear Creek between Clear Creek Road and the Sacramento River. Report prepared by Sacramento Fish and Wildlife Office, USFWS, Sacramento, CA.

Valdimarsson, S. K., N. B. Metcalfe, J. E. Thorpe, and F. A. Huntingford. 1997. Seasonal changes in sheltering: Effect of light and temperature on diel activity in juvenile salmon. Animal Behavior **54**:1405-1412.

Van Winkle, W., H. I. Jager, and B. D. Holcomb. 1996. An individual-based instream flow model for coexisting populations of brown and rainbow trout. EPRI TR-106258, Electric Power Research Institute, Palo Alto, CA.

Van Winkle, W., H. I. Jager, S. F. Railsback, B. D. Holcomb, T. K. Studley, and J. E. Baldrige. 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.

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**: 1293-1297.

Wilcock, P. R.; Barta, A. F.; Shea, C. C.; Kondolf, G. M.; Matthews, W. V. G.; Pitlick, J. 1996. Observations of flow and sediment entrainment on a large gravel-bed river. Water Resources Research: **32**: 2897-2909.

Appendix B: Software Guide

inSALMO Version 1.0 Software Guide

Prepared by:

Steven F. Railsback
Lang, Railsback & Associates
Arcata, CA
www.LangRailsback.com

Colin Sheppard
Arcata, CA

Last changed: October 10, 2011

Contents

List of Figures **6**

List of Tables **7**

1. Overview of the *inSALMO* Software Help File **8**

2. General Information on Input Files **8**
2.1. Common Characteristics of Input Files 8
2.2. Variable types 9
2.3. Files read by the Swarm object loader 9
2.1. Color variables 10
2.2. Translating files between Unix and Windows 10

3. Setup Files **10**
3.1. Observer Setup 10
3.2. Species Setup 12
3.3. Reach Setup 13
3.4. Model Setup 15
3.5. Experiment Setup 17

4. Parameter Files **17**
4.1. Habitat parameter file 18
4.2. Salmon parameter file 18

5. Data Files **22**
5.1. Cell geometry 22
5.2. Cell data 23
5.3. Hydraulic data 23
5.4. Spawner initialization data 25
5.5. Flow, temperature, and turbidity time series data 26

6. Changing Species and Size Classes **28**
6.1. Adding or Removing Species 28
6.2. Changing Size Classes 30

7. Graphical Interfaces **33**
7.1. Main Control Panel 33
7.2. Model Run Controller 34
7.3. Habitat Probe Displays 35
7.4. Graphs 36
7.5. Animation Window 36
7.5.1. Probe displays 37
7.5.2. Movies of the animation window 40

8. File Output **41**
8.1. General Information on Output Files 41
8.2. Output File Descriptions 42

9. Optional Output Files for Testing and Specialized Studies **43**

10. Experiment Manager **47**
10.1. What the Experiment Manager Does 47

10.2. General Procedure for Setting Up Experiments 47
10.2.1. Experiment.Setup format 48
10.2.2. Class and instance names for typical experiments 49
10.2.3. Valid parameter value types 50
10.2.4. Using instance names 51
10.2.5. Controlling where output goes 53
10.2.6. Checking the Experiment Manager 53
10.3. Example Experiment.Setup Files 54
10.3.1. No experiment 54
10.3.2. Replicate simulations 54
10.3.3. Parameter sweep for salmon 55
10.3.4. Habitat parameter sweep 55
10.3.5. Multiple parameter sweep 56
10.3.6. Alternative daily input files 57
10.3.7. Multiple simulation years 58
10.3.8. Alternative cell data files 58

11. Trouble-shooting Guide 60

List of Figures

Figure 1. Example Observer.Setup file. 11

Figure 2. Example Species.Setup file. 12

Figure 3. Example Reach.Setup file. 14

Figure 4. Example Model.Setup file. 16

Figure 5. Example habitat parameter file. 18

Figure 6. Example salmon parameter file. 21

Figure 7. Example cell geometry file. 22

Figure 8. Example cell data file. 23

Figure 9. Example hydraulic data file. 24

Figure 10. Example spawner initialization data file. This file defines spawners for two reaches and three years. 25

Figure 11. Example file for flow, temperature, and turbidity input. 27

Figure 12. Main control panel. 33

Figure 13. Model Run Controller. 34

Figure 14. Habitat probe display. One of these windows is opened for each habitat reach. 35

Figure 15. Mortality output graph. “Time” on the X axis is the number of days simulated. The Y axis is the total number of salmon that have died of each of the seven mortality sources. 36

Figure 16. Animation window. One such window is opened for each habitat reach, with the reach name used as the window’s title (top left). 37

Figure 17. Cell probe display. 38

Figure 18. Salmon and redd probe displays. 39

Figure 19. Right-clicking on the species name box in a salmon probe display opens a (empty) complete probe display (left). Clicking on the green button opens a complete probe display for the salmon (right). 40

Figure 20. Example Experiment.Setup file. 49

List of Tables

Table 1. Contents of Observer.Setup. 11

Table 2. Contents of the Reach.Setup file. 15

Table 3. Contents of the Model.Setup file.....17

Table 4. Optional output files.....45

Table 5. Class and instance names for parameters commonly used in the Experiment Manager. 50

Table 6. Value types for the Experiment.Setup file. 51

Table 7. Trouble-shooting guide.....60

1. Overview of the *inSALMO* Software Help File

This digital document provides user help on how to assemble input for the *inSALMO* version 1.0 salmon model, how to set up and control model runs, and on the kinds of output the model produces.

This document does *not* describe the graphical user interface (GUI) that can optionally be used to create and modify input files and run the model. Documentation for the GUI is provided in a separate help file. Instead, this document explains the files that the model software itself reads as input and creates as output. The GUI can be used to modify the input files or users can edit them directly.

2. General Information on Input Files

The following three sections provide the detailed format for all the files needed to define an *inSALMO* run. Setup files provide run-control information, parameter files provide equation coefficients for salmon and habitat, and data files provide the numbers needed to define habitat cell characteristics, the initial salmon populations, and the time series of flow, temperature, and turbidity that drive the model.

2.1. Common Characteristics of Input Files

All of the setup, parameter, and data input files are in ASCII. They can be maintained and edited using ASCII editors (e.g., Notepad, WordPad, gvim), or by using word processor or spreadsheet software and saving them as ASCII (plain text) or (as noted below) in .CSV format. Spreadsheets are especially useful for maintaining some input files.

Because *inSALMO* runs in a Unix-like environment (even under Windows), it uses case sensitive file names and variable names. The model needs a file named “Model.Setup” and it will not find and use files named “model.setup” or “Model.setup”. Similarly, a variable named “fishParam” is different from one named “fishparam”. Failing to notice case differences in file and variable names is a common source of frustration.

None of the files require values to be in any particular columns: blanks in the file are ignored (with a few exceptions that are carefully noted). Values on the same line can be separated by one or more space or tab characters.

Many of the input files start with three header lines that are ignored by the software; these can be used to document the file type and where its information came from, and to provide column labels for the remaining lines. These header lines can be up to 200 characters long.

A new feature in version 1.0 of *inSALMO* is the ability to read most data files (sections 5.2-5.5) in comma-separated variable (CSV) format. This format is widely used for input and output from spreadsheet, statistical, and other data-oriented programs. These files can be generated, for example, by saving Excel spreadsheets in CSV format. CSV files are in ASCII (“plain text”) with commas separating the variables in a file line.

inSALMO does not completely implement the CSV format standard. According to the CSV standard, it is allowable (but not required) for any inputs, including numbers, to be inclosed in double quotes, but input for *inSALMO* (except header information) must not be in quotes. Excel

generally does not put numbers or text values in quotes when spreadsheets are saved in .CSV format, so usually this issue is unimportant for Excel users. The Calc spreadsheet of OpenOffice and LibreOffice does however enclose text values in quotes, which must then be stripped out using a text editor.

2.2. Variable types

This document refers to variables as belonging to several common types:

- *Integers* are numbers with no decimal places and often describe how many there are of some thing (e.g., how many fish at the start of a simulation).
- *Floats* are numbers with decimal places (what mathematicians call "real numbers"). They can be written in decimal format (e.g., `0.074`) or scientific notation (`7.4E-2`).
- *Text* are variables containing words or strings of alphanumeric characters. Text variables are case-dependent and cannot include blanks. They should not be enclosed in quotes.
- *Dates* are in MM/DD/YYYY format (e.g., `10/1/2008`)

2.3. Files read by the Swarm object loader

The setup files and parameter files are read by Swarm's "object loader" facility. The object loader is a simple tool for reading in variable values for an object. Except as noted below, all the setup and parameter files must be in the following object loader format.

- The first line has only the text `@begin` (unless the first lines are comments; see below).
- There is one line per variable, with each line containing the variable name followed by its value.
- Variables need not be in any particular order.
- The variable must be spelled <u>exactly</u> as it is in the code, including upper/lower case.
- For text variables, the value text does <u>not</u> use quotation marks.
- Variables containing text <u>must not have trailing blanks after the text input</u>. For example, a variable that has values of either `NO` or `YES` must not have a blank space after the value. (The blank will be read as part of the variable's value, which makes the code unable to interpret the value. If the variable is a file name, the code will be unable to open the file because it will look for a file with a blank at the end of its name. This is another potential source of frustration.)
- Integer values should not have a decimal point.
- The last line has only the text `@end`.

- Comment lines (ignored by the computer) can be included in the file; they start with the character #. (In a few strange cases, Swarm has been unable to distinguish comments so they had to be removed.)

Example object loader files are provided in the following sections.

2.1. Color variables

Several setup files specify color variables: the color used to display various objects. Color variables contain a text word that is the color name (e.g., `bluegreen`). Allowable values include all the common colors (red, green, blue, etc.) plus a large number of exotic color names. (Lists of Unix colors can be found on-line; search for "Unix colors".) Invalid color names result in white being used. Do not use color names that contain blanks: e.g., use `DarkViolet` instead of the equivalent `dark violet`.

2.2. Translating files between Unix and Windows

(Users who work only in Windows can ignore this section.) Transferring files between Windows and Unix-based (including Linux) operating systems can result in subtle problems because the two operating systems use different codes for the line ends in ASCII files. Hence, files created in Windows may not work in Linux; and files created in Linux may work in some, but not all, Windows programs. Attempting to run *inSALMO* in Unix with files created (or edited) in Windows is likely to fail because of this problem (and Unix will not clearly indicate what the problem is).

Unix and Linux operating systems include programs "dos2unix" and "unix2dos" to convert between formats. For example, to convert a directory of files prepared in Windows for use on a Linux computer, in Linux type `dos2unix *` within the directory. Be aware, though, that <u>dos2unix and unix2dos destroy any executable file</u> they attempt to convert, so be careful using them on directories that contain *inSALMO*.exe.

3. Setup Files

Five setup files provide run-time control information about graphical outputs, species, habitat reaches, automated experiments, and the model run itself.

3.1. Observer Setup

This file controls the observer swarm, which provides the graphical interfaces described in Section 7. It must be named "Observer.Setup". An example is at Figure 1.

```
@begin
rasterColorVariable         depth
takeRasterPictures          NO

rasterResolutionX           50
rasterResolutionY           50
maxShadeVelocity            200
maxShadeDepth               150

tagFishColor                 tomato
tagCellColor                 DeepPink1
dryCellColor                 white
@end
```

Figure 1. Example Observer.Setup file.

The variables in Observer.Setup are explained in Table 1.

Table 1. Contents of Observer.Setup.

Variable name and type	Definition
rasterColorVariable (text)	Selects the habitat variable used to color-code habitat cells on the animation window. Valid values are `depth` and `velocity`.
takeRasterPictures (text)	Determines whether the raster window is captured in files for post-processing into a movie of the simulation (see Section 7.5.2). Valid values are no and `yes` (also `NO` and `YES`). Raster pictures should not be turned on unnecessarily; they severely reduce execution speed and generate many large output files.
rasterResolutionX (integer) *rasterResolutionY* (integer)	Set the animation window resolution (cm per display pixel) in the X (east-west) and Y (north-south) dimensions. Reducing these variables makes the displays bigger, but requires more memory and more execution time to update the display.The values of *rasterResolutionX* and *rasterResolutionY* can be varied separately to exaggerate one dimension.
maxShadeVelocity (float) *maxShadeDepth* (float)	Control the animation window's shading by depth or velocity. These variables are the maximum velocity (depth) over which cells are shaded; velocities (depths) greater than these values are given the same color. For example, if *maxShadeVelocity* is 200, then cells will be shaded from yellow to red as velocity increases from 0 to 200 cm/s; but all cells with velocity > 200 will have the same, reddest, color. Increasing these variables makes it possible to distinguish

	among higher velocities (depths) but harder to distinguish among low values.
tagFishColor (text) *tagCellColor* (text)	Set the color for fish and cells that have been tagged via probes. (See Section 2.1 about color variables.)
dryCellColor (text)	Sets the display color of cells with zero depth.

3.2. Species Setup

The species setup file tells *inSALMO* which species are being simulated and where to find the input files for each. “Species” can refer to different races or runs of salmon as well as different species. (This file does not, by itself, determine how many species are modeled; see Section 6.1.) The file must be called “Species.Setup”. The contents of this file must match the code that defines species in the model, as explained in Section 6.1.

The species setup file does not use the Swarm object loader format. A Species.Setup example for a two-species model is at Figure 2.

```
Species.Setup file, fall and spring Chinook model.
For each species, provide species class name, parameter file name,
spawner initialization file name, and raster display color.

FallChinook
TurbidCrkFallChinook.Params
TurbidCrkFallSpawnerInitFile.csv
black

SpringChinook
TurbidCrkSpringChinook.Params
TurbidCrkSpringSpawnerInitFile.csv
olivedrab
```

Figure 2. Example Species.Setup file.

The species setup file starts with a block of three comment lines that are ignored by the software, followed by a blank line. Then come a block of lines for each species in the model. Each such block includes:

- A line with the species name, which must exactly match the name of the source code class for the species. (If this file contains a species “FallChinook”, then the source code files FallChinook.h and FallChinook.m must have been compiled when *inSALMO*.exe was created.)

- A line with the name of the fish parameter file that the species uses. (One parameter file can be used by more than one species.)
- A line with the name of the spawner initialization data file for the species.
- A line containing the name of the color used for the species in the animation window.
- A blank line, if another species follows.

3.3. Reach Setup

This setup file, which must be named “Reach.Setup”, specifies the number of habitat reaches, how reaches are linked, and what input files should be used for each reach. The file does <u>not</u> use the object loader format. The file contains:

- Three header lines that are ignored by the computer.
- A blank line
- A block of lines for each reach. These blocks start with a line containing only the word `REACHBEGIN` and end with a line containing only the word `REACHEND`. There can be multiple blank lines between these blocks.

```
Reach.Setup file.
Provide one block for each reach.
Example input.

REACHBEGIN
reachName                        WeejakTrib
habParamFile                     LJCHab.Params

habDownstreamJunctionNumber       2
habUpstreamJunctionNumber         3

cellGeomFile                     LJCWeejGeom.Data
cellHabVarsFile                  LJCWeejCell.Data
cellHydraulicFile                LJCWeejHydr.Data
flowFile                         WeejTestFlow.Data
temperatureFile                  LJCLowTemp.Data
turbidityFile                    LJCLowTurbidity.Data

REACHEND

REACHBEGIN
reachName                        LowerMainstem
habParamFile                     LJCHab.Params

habDownstreamJunctionNumber       4
habUpstreamJunctionNumber         2

cellGeomFile                     LowerMainstemGeom.Data
cellHabVarsFile                  LowerMainstemCell.Data
cellHydraulicFile                LowerMainstemHydr.Data
flowFile                         LowerMainstemFlow.Data
temperatureFile                  LJCLowTemp.Data
turbidityFile                    LJCLowTurbidity.Data

REACHEND
(etc. for remaining reaches)
```

Figure 3. Example Reach.Setup file.

The block of lines for each reach contains a separate line for each of the reach's setup variables. The line contains the variable name, one or more spaces, then the variable value. These lines need not be in any particular order within the block. An example Reach.Setup file is at Figure 3, with variables explained in Table 2.

Table 2. Contents of the Reach.Setup file.

Reach variable and type	Definition
reachName (text)	The user-defined name for a reach. (Up to 30 characters.)
habParamFile (text)	The name of the reach's habitat parameter file.
habDownstream-JunctionNumber (integer)	The junction number for the reach's downstream end. (Junction numbers are explained in the model description document.)
habUpstreamJunction-Number (integer)	The junction number for the reach's upstream end.
cellGeomFile (text)	The name of the reach's cell geometry file. File names can be up to 35 characters long.
cellHabVarsFile (text)	The name of the reach's cell habitat variables file.
cellHydraulicFile (text)	The name of the reach's hydraulic input file.
flowFile (text) *temperatureFile* *turbidityFile*	The names of the flow, temperature, and turbidity data files for the reach to use. (Different reaches can use the same files.)

3.4. Model Setup

The model setup file must be called "Model.Setup". It contains basic variables controlling a model run. An example is at Figure 4, and the file contents are explained in Table 3.

```
# Model Setup File for Example Salmon Model
# Created 8/04/2010

@begin

randGenSeed                        32461
numberOfSpecies                        1

runStartDate                     10/1/1990
runEndDate                       9/30/2001
fishOutputFile               LiveFish.out
fishMortalityFile            DeadFish.out
reddOutputFile               Redds.out

fileOutputFrequency                    10
appendFiles                   0

siteLatitude                   42

@end
```

Figure 4. Example Model.Setup file.

Table 3. Contents of the Model.Setup file.

Reach variable and type	Definition
randGenSeed (integer)	The seed value for the random number generator used for all stochastic processes except year shuffling. It can be any positive integer.
numberOfSpecies (integer)	The number of species to simulate. The number must be no higher than the number of species defined in the species setup file. For example, if 3 species are defined in Species.Setup but *numberOfSpecies* is 2, then the first two species in Species.Setup are simulated. No more than 10 species can be modeled.
runStartDate *runEndDate* (date: mm/dd/yyyy)	The dates for which simulations begin and end.
fishOutputFile (text)	The name of the output file for statistics on live fish.
fishMortalityFile (text)	The name of the output file for statistics on fish mortality.
reddOutputFile (text)	The name of the output file for statistics on redds.
fileOutputFrequency (integer)	The frequency with which file output is written. If set to 1, output is written for each simulated day; if set (for example) to 10, output is written only each 10th day. (This output is the model's state on the date when output is written, not an average for the period between output dates.)
appendFiles (boolean: yes or no)	Whether existing output files should be appended (instead of over-written) at the start of each model run. (See Section 10 concerning this variable and the Experiment manager.) Valid values are `1` for yes and `0` for no.
siteLatitude (floating point)	The latitude of the study site, in degrees north (used to calculate day lengths).

3.5. Experiment Setup

The experiment setup file "Experiment.Setup" controls *inSALMO*'s experiment manager. This feature is complex enough that Section 10 is dedicated to it.

Users are reminded to inspect Experiment.Setup any time *inSALMO* appears to be ignoring a setup variable or parameter (another common source of frustration).

4. Parameter Files

inSALMO requires two kinds of parameter files, for habitat and salmon. These files provide the values for all the parameters defined in the model description document. Each habitat reach,

and each salmon species, must have a parameter file assigned to it (this assignment is done in the reach and species setup files). More than one reach, or species, can use the same parameter file. Parameter files use the Swarm object loader format (Section 2.3).

4.1. Habitat parameter file

The name of the habitat parameter file for a reach is defined by the user in the reach's setup file (Section 3.3). By convention, the file name includes the reach name, the syllable "Hab", and the extension ".Params"; for example, "SmithCrkMiddleReachHab.Params".

Figure 5 provides an example habitat parameter file. Habitat parameter files must include exactly the same variables as in this example. However, the order in which parameters appear does not matter.

```
@begin
habSearchProd                  7.0E-7
habDriftConc                   1.50E-10
habDriftRegenDist                   500
habPreyEnergyDensity               2500
habMaxSpawnFlow                    4.0
habShearParamA                    0.019
habShearParamB                    0.383
habShelterSpeedFrac               0.3
@end
```

Figure 5. Example habitat parameter file.

4.2. Salmon parameter file

The user defines the name of the parameter file for each salmon species in the species setup file (Section 3.2). By convention, the file name includes the species name, perhaps the site name, and the extension ".Params"; an example is "SmithCrkFallChinook.Params".

Figure 6 provides an example salmon parameter file. All salmon parameter files must include exactly the same variables as in this example; the parameters are defined in the model description document. (This figure does not necessarily include the "standard" or most up-to-date parameter values for any site.) The software checks to make sure all the salmon parameters are initialized: if any salmon parameters are missing from this file, an error statement will be issued and execution will stop.

```
@begin

fishCaptureParam1                  1.6
fishCaptureParam9                  0.5

fishCmaxParamA                   0.628
fishCmaxParamB                    -0.3
fishCmaxTempF1                    0.05
fishCmaxTempF2                    0.05
fishCmaxTempF3                     0.5
fishCmaxTempF4                       1
fishCmaxTempF5                     0.8
fishCmaxTempF6                       0
fishCmaxTempF7                       0
fishCmaxTempT1                       0
fishCmaxTempT2                       2
fishCmaxTempT3                      10
fishCmaxTempT4                      22
fishCmaxTempT5                      23
fishCmaxTempT6                      25
fishCmaxTempT7                     100

fishDetectDistParamA                 4.0
fishDetectDistParamB                 2.0
fishEnergyDensity                 5900

fishFecundParamA                  690
fishFecundParamB                  0.552

fishFitnessHorizon                  90

fishMaxSwimParamA                  2.8
fishMaxSwimParamB                   21
fishMaxSwimParamC               -0.0029
fishMaxSwimParamD                0.084
fishMaxSwimParamE                 0.37
fishMoveDistParamA                 100
fishMoveDistParamB                   2

fishOutmigrateSuccessL1            5.0
fishOutmigrateSuccessL9           12.0

fishRespParamA                      30
fishRespParamB                   0.784
fishRespParamC                  0.0693
fishRespParamD                    0.03
fishSearchArea                   20000

fishSpawnEggViability              0.8
fishSpawnDefenseArea            200000

fishSpawnStartDate                10/1
fishSpawnEndDate                 11/30

fishSpawnDSuitD1                   0.0
fishSpawnDSuitD2                  12.0
fishSpawnDSuitD3                  27.0
```

```
fishSpawnDSuitD4                  33.5
fishSpawnDSuitD5                 204.0
fishSpawnDSuitS1                   0.0
fishSpawnDSuitS2                   0.0
fishSpawnDSuitS3                  0.95
fishSpawnDSuitS4                   1.0
fishSpawnDSuitS5                   0.0
fishSpawnMaxFlowChange             0.2
fishSpawnMaxTemp                    14
fishSpawnMinTemp                     5
fishSpawnProb                      0.2
fishSpawnVSuitS1                   0.0
fishSpawnVSuitS2                   0.0
fishSpawnVSuitS3                  0.06
fishSpawnVSuitS4                   1.0
fishSpawnVSuitS5                   1.0
fishSpawnVSuitS6                   0.0
fishSpawnVSuitV1                   0.0
fishSpawnVSuitV2                   2.3
fishSpawnVSuitV3                   3.0
fishSpawnVSuitV4                  54.0
fishSpawnVSuitV5                  61.0
fishSpawnVSuitV6                 192.0
fishSpawnWtLossFraction            0.4

fishTurbidExp                  -0.0711
fishTurbidMin                      0.1
fishTurbidThreshold                5.0

fishWeightParamA               0.00411
fishWeightParamB                  3.49

mortFishAqPredD1                    20
mortFishAqPredD9                    10
mortFishAqPredF1                    18
mortFishAqPredF9                     0
mortFishAqPredL1                     4
mortFishAqPredL9                    18

mortFishAqPredMin                 0.92

mortFishAqPredT1                    15
mortFishAqPredT9                     8
mortFishAqPredU1                     5
mortFishAqPredU9                    80
mortFishConditionK1                0.3
mortFishConditionK9                0.6
mortFishHiTT1                     28.0
mortFishHiTT9                     24.0
mortFishStrandD1                  -0.3
mortFishStrandD9                   0.3
mortFishTerrPredD1                   5
mortFishTerrPredD9                 200
mortFishTerrPredF1                  18
mortFishTerrPredF9                   0
mortFishTerrPredH1                 500
mortFishTerrPredH9                -100
mortFishTerrPredL1                   6
mortFishTerrPredL9                   3
```

```
mortFishTerrPredMin               0.95

mortFishTerrPredT1                  10
mortFishTerrPredT9                  50
mortFishTerrPredV1                  20
mortFishTerrPredV9                 200
mortFishVelocityV1                 1.8
mortFishVelocityV9                 1.4

mortReddDewaterSurv                0.9
mortReddHiTT1                     23.0
mortReddHiTT9                     17.5
mortReddLoTT1                      1.7
mortReddLoTT9                      4.0
mortReddScourDepth             10000.0

reddDevelParamA                33000.0
reddDevelParamB                  -2.04
reddDevelParamC                  -7.58
reddNewLengthMin                   3.5
reddNewLengthMax                   4.1
reddSize                         56000

@end
```

Figure 6. Example salmon parameter file.

5. Data Files

Five kinds of files are used to define habitat and the initial salmon population.

5.1. Cell geometry

There is one geometry file per reach, providing the coordinates of the corners of each habitat cell. Cells are polygons that can have any number of sides. Each cell is designated by a cell number that can be arbitrary, but each cell's number must be unique (in its reach; cells in different reaches can have the same number). The geometry can be in any coordinate system, being treated by *inSALMO* as plain Euclidean coordinates. Different reaches can use different coordinate systems with different origins; coordinate are transformed by *inSALMO* into an internal spatial representation that does not consider the actual distance among reaches. However, *inSALMO* follows the conventions of UTM coordinates, with units of meters and a Mercator projection that assumes the x direction is east and y is north.

The geometry file format is adopted from the "ungenerate" command of the ArcGIS geographic information system (other GIS packages can also use this format). "Ungenerate" exports the geometry of a polygon system as an ASCII text file. The file format is described in ArcGIS documentation and illustrated by Figure 7. This example depicts three cells, numbers 1, 3, and 4. Each cell is described by a block of data lines, with values separated by spaces (the spacing within a line is not important).

The first line for a cell contains the cell number, followed by x and y coordinates of a cell label that is ignored by *inSALMO*. Following lines provide the x and y coordinates of each corner of the cell. A line containing just "END" delineates the end of each cell, and two such lines indicate the end of input.

```
         1      2923.3222656      2084.7429199
      2917.0759277      2079.6772461
      2926.8261719      2089.6940918
      2929.4162598      2089.6560059
      2918.4470215      2078.3823242
      2917.0759277      2079.6772461
END
         3      2936.2717285      2087.4851074
      2929.4162598      2089.6560059
      2942.2133789      2087.4470215
      2942.0610352      2084.7048340
      2930.1779785      2087.5612793
      2929.4162598      2089.6560059
END
         4      2937.0334473      2081.5056152
      2942.0610352      2084.7048340
      2941.4516602      2076.0590820
      2934.9006348      2073.7739258
      2930.1779785      2087.5612793
      2942.0610352      2084.7048340
END
END
```

Figure 7. Example cell geometry file.

5.2. Cell data

There is one cell data file for each reach modeled; it provides the static habitat variable values for each cell. The name of a reach's cell data file is specified by the user in the reach setup file. The convention is for these file names to include the reach name and end in "Cell.Data" (e.g., SmithCrkLowerReachCell.Data).

The file still starts with three rows of header information, followed by one row for each cell in the reach. Each of these cell rows contains only four values, separated by spaces, tabs, or commas. The values are:

- The cell number, using the same numbering as in the cell geometry file (Section 5.1). Cells need not appear in any particular order, and cell numbers need not be sequential.
- The fraction of the cell providing velocity shelter for drift-feeding juvenile salmon.
- A characteristic distance to hiding cover for the cell (in meters, not cm).
- The fraction of the cell providing suitable spawning gravel.
- A code for which end of the reach the cell is at. Valid values are "U" if the cell is in the main channel at the reach's upstream end, "D" if the cell is in the main channel at the downstream end, and "I" for all other cells. These codes are used by fish to calculate how far they are from their reach's ends.

Figure 8 is a (partial) example cell data file. The values in this file can be separated by spaces or tabs, or the file can be in CSV format (Section 2.1; illustrated in Figure 8).

```
Turbid Creek Upstream site cell habitat variables file
Last modified: SFR 4/20/2011
Cell#,FracVelShelter,DistToHidingCover,FracSpawnGravel,ReachEndCode
2,0.1,0.5,0.0,U
4,0.0,2.2,0.0,I
5,0.1,1.5,0.0,U
6,0.1,1.5,0.0,U
7,0.1,1.5,0.5,U
...
```

Figure 8. Example cell data file.

5.3. Hydraulic data

To model how the depth and velocity of each cell varies with flow, *inSALMO* imports the depth and velocity of each cell at each of many flows (explained in the model description). This information is in one hydraulic data file per reach. The names of these files are provided by the user, for each habitat reach, in the reach setup file (Section 3.3). The naming convention for hydraulic data files is to include the reach name and end with "Hyd.Data", e.g., SmithCrkHyd.Data.

The hydraulic data file format was designed assuming that its contents would be prepared using geographic information system (GIS) software and analysis. This design allows output from any hydraulic model (including one-dimensional models) to be used in *inSALMO* by importing

hydraulic model geometry and results to GIS, processing and checking it as needed, and saving the information in the format used by *inSALMO*. Another reason for using GIS to process hydraulic data is that it allows the hydraulic model to use a spatial resolution and grid mesh different from the cells used in *inSALMO*. As the Clear Creek example (Sect. 8.2) shows, results from a highly detailed hydraulic simulation can be processed in GIS to produce average depths and velocities for the larger *inSALMO* cells. The depths and velocities should be depth-averaged values that represent the entire cell area.

Some hydraulic models assign a negative depth to cells that are above water, so *inSALMO* converts any negative depths to zero when they are read in. However, any negative velocities are treated as an error because input should be the velocity magnitude, which cannot be negative—hence, negative velocity input likely indicates that a velocity component is mistakenly being used instead of the magnitude. If the model encounters a negative velocity it raises an error statement and stops.

The hydraulic data file for each reach (illustrated at Figure 9) includes:

- Two lines of header information ignored by the computer.
- A line that starts with the word “Flows:”, and then lists all the flows in the lookup table. The flows must be in ascending order.
- Another line of header information that should be used to label the following columns by which flow they represent and whether they contain depth or velocity (in the example below, “D@1.42” refers to depth at 1.42 m^3/s, etc.).
- One line for each cell, containing the cell number, then the depth followed by velocity for each flow.

This file can also be in CSV format.

```
Hydraulic data input file for inSALMO 1.0
Clear Creek Reach 3a Data 04 OCT 2010 by D.S. Montoya
Flows: 1.42         2.12          2.83
Cell D@1.42 V@1.42 D@2.12 V@2.12 D@2.83 V@2.83
2       -0.73 0.00   -0.70  0.00   -0.67  0.00
4       -0.70 0.00   -0.67  0.00   -0.64  0.00
5       -0.72 0.00   -0.68  0.00   -0.65  0.00
6       -0.60 0.00   -0.56  0.00   -0.53  0.00
7       0.77  0.07   0.80   0.10   0.83   0.12
8       1.15  0.13   1.18   0.19   1.21   0.24
10      0.96  0.12   0.99   0.17   1.02   0.22
...
```

Figure 9. Example hydraulic data file.

The software makes several checks when reading in the cell hydraulic file:

- The cell numbers must correspond with those in the cell geometry file: each cell defined in the geometry file receives hydraulic input and there are no cells in the hydraulic file but not in the geometry file.

- The flows on line 3 must be in order of increasing magnitude.
- The number of values on each row of hydraulic input (file lines 5 and higher) must equal one (the cell number) plus two times the number of flows defined on line 3.
- There must be no negative velocities.
- Any negative depth values are set to zero (some hydraulic models assign negative depths to locations above the water level).

5.4. Spawner initialization data

Spawner initialization data specifies the number, characteristics, and timing of adult salmon arriving to spawn. This input is provided in a separate file for each species. Each such file includes data for all reaches and all years. The file name convention is to include the species and study site names, and end in "SpawnerInit.Data" (e.g., SmithCrkFallChinookSpawnerInit.Data).

The file starts with three comment lines that are ignored by the software; the third line is usually column headings. Next come lines of data that each specify the initial number and size of salmon of one age, for one reach. This file can be in CSV format, so can be maintained in a spreadsheet that looks like Figure 10.

Spawner Initialization file	Values estimated from CDF&G carcass counts	JWhite	July 2011					
Fall Chinook in Clear Creek			Save in .csv format					
Year	Arrival reach	Number of spawners	Fraction female	Arrival start date	Arrival end date	Arrival ratio	Female length mean	Female length SD
1998	ClearCreek-3A	324	0.58	10/2/1998	10/31/1998	0.0001	85	8.1
1999	ClearCreek-3A	580	0.51	10/2/1999	10/31/1999	0.0001	85	8.1
2000	ClearCreek-3A	485	0.5	10/2/2000	10/31/2000	0.0001	85	8.1
1998	ClearCreek-3C	304	0.58	10/2/1998	10/31/1998	0.0001	85	8.1
1999	ClearCreek-3C	545	0.51	10/2/1999	10/31/1999	0.0001	85	8.1
2000	ClearCreek-3C	455	0.5	10/2/2000	10/31/2000	0.0001	85	8.1

Figure 10. Example spawner initialization data file. This file defines spawners for two reaches and three years.

Each data line (line 4 and higher) includes the following values:

- Initialization year. This value is only for the user's convenience.
- Name of the reach where the spawners will "arrive" and spawn.
- Number of adults that will be created.

- Arrival start date: the first day on which spawners arrive.
- Arrival end date: the last day on which spawners arrive.
- Arrival ratio, the parameter that controls how peaked the distribution of arrivals is between the start and end dates.
- The mean length (cm) of the spawners. This value is used only to determine fecundity, so should represent females, not males.
- The standard deviation in length (cm) of the initial spawners, also for females only.

5.5. Flow, temperature, and turbidity time series data

inSALMO uses daily input values of flow, temperature, and turbidity, for each reach. Each of these variables is provided in a separate file (each data file provides one time series of one variable). The names of the flow, temperature, and turbidity data files used by each reach is specified by the user in the reach setup file. The file names usually include the reach name, the type of data, and ".Data" (e.g., SmithCrkFlow.Data, SmithCrkTurbid.Data).

These data files are read by the EcoSwarm class TimeSeriesInputManager. The format for each is:

- The first three lines are headers ignored by the software. The third line usually provides column headings.
- Remaining lines provide the value for one day. Each line starts with the date (mm/dd/yyyy), followed by the daily value.

The data lines must be in increasing order by date, and no days (including leap days) may be omitted. Flow values are in m^3/s, temperatures in °C, and turbidity in NTU. Figure 11 is an example file. CSV format is allowed.

```
Temperature data for LJC Lower Site. From 1999-00 measured values
Assembled 1/11/01
Date  Temperature (C)
10/1/1987   11.6
10/2/1987   11.6
10/3/1987   11.5
10/4/1987   11.4
10/5/1987   11.4
10/6/1987   11.3
10/7/1987   11.3
10/8/1987   11.2
10/9/1987   11.1
10/10/1987  11
10/11/1987  10.9
10/12/1987  10.7
...
```

Figure 11. Example file for flow, temperature, and turbidity input.

These files may include dates outside the range being simulated in any particular model run; *inSALMO* simply finds the values it needs from within the data files. Flow data must include the day after the last simulation day (for modeling redd scouring mortality).

Spreadsheet software is especially convenient for building these time series data files. However, when data are imported to a spreadsheet it can convert the dates to a different format (e.g., dates show up in the spreadsheet as “10/1/87” instead of “10/1/1987”, a common source of frustration). It can therefore be necessary to re-format the date column in the spreadsheet before saving it as the ASCII or CSV file used by *inSALMO*. Another frequent problem is spreadsheet columns that are slightly too narrow, so some dates (e.g., 10/15/1987 but not 1/5/1987) are saved as “#######”. (Using CSV format to save the file for input to *inSALMO* avoids this problem.) Parsing errors also commonly occur because the number of characters in a date changes. If the date column starts at 1/1/2001, the spreadsheet may decide that the date column needs to be only 8 characters wide. When dates like 10/10/2001 are reached, the last characters end up in the second spreadsheet column, ruining the file. Always inspect spreadsheets carefully before saving the data for input to *inSALMO*.

6. Changing Species and Size Classes

Minor changes to the software's source code are needed to change either the species represented by *inSALMO* or the size classes used to summarize and report output (Section 8). These changes do not require programming skills if the following directions are followed, but it is recommended that users save their original source code separately before making the changes.

6.1. Adding or Removing Species

inSALMO can use up to 10 species or races of salmon. The number of species included in a run is set in the Model.Setup file, and the files used by each species are specified in the Species.Setup file. However, these setup files can only refer to species that exist in the software as a subclass of the "Trout" class. To exist in the software, a species must have its own interface (.h) file and implementation (.m) file. Therefore, adding a species is simply a matter of creating an interface and implementation file for the species; and telling the software to include those files when it is compiled, by including the new file names in the Makefile (explained below). Likewise, species can be removed by deleting their files and removing reference to them from the Makefile. And one species can be replaced by another by simply replacing the species name in all the locations discussed here.

It is critical that the name of the new species' interface and implementation files exactly match the species' name in the Species.Setup file. The new .h and .m files must of course be in the same directory as the rest of the source code.

The following steps add a species to *inSALMO*. As an example, the rare blarney salmon is added to a model already containing fall and spring Chinook salmon. If any of the six changes are incomplete, the software may exit with an error or crash without explanation.

Step 1. Create the species interface file. This is easiest done by copying and editing the .h file of an existing species. For example, the file "FallChinook.h" can be copied to a new file "Blarney.h" and edited to:

```
#import "Trout.h"
@interface Blarney : Trout
{

}
+ createBegin: aZone;

@end
```

Step 2. Create the species implementation file. Again, this is easiest done by copying and editing the .m file of an existing species. The new "Blarney.m" file should be edited to:

```
#import "globals.h"
#import "Blarney.h"

@implementation Blarney

+ createBegin: aZone
{
   return [super createBegin: aZone];
}

@end
```

Step 3. Add the new species to the TroutModelSwarm.h file. Near the top of TroutModelSwarm.h is a series of “#import” statements that say which other .h files are used by TroutModelSwarm.h. Simply add a new #import statement for the new species, just like the statement for existing species:

```
#import <stdlib.h>
#import <objectbase/Swarm.h>
#import <analysis/Averager.h>

#import "TroutModelSwarmP.h"

#import "globals.h"
#import "FallChinook.h"
#import "SpringChinook.h"
#import "Blarney.h"  // New species added here
#import "Redd.h"
#import "HabitatSpace.h"
#import "FishParams.h"
```

Step 4. Modify the Species.Setup file by adding lines for the new species (see Section 3.2).

Step 5. Modify the Model.Setup file to increase the parameter *numberOfSpecies*. The value of this parameter must not be greater than the number of species listed in Species.Setup.

Step 6. Edit the Makefile in three places. In the source code directory is a file called “Makefile”, which provides the compiler with directions for which source code files to include in *inSALMO*. Users need not try to understand the Makefile, but can just add the new species wherever an existing species is found. First, the OBJECTS statement must have the new species’ class file in it. This statement should look like (your makefile may not look exactly like this):

```
OBJECTS=Trout.o Cell.o Vector.o HabitatSpace.o \
      Redd.o \
      TroutModelSwarm.o \
      TroutObserverSwarm.o \
…

\
        FallChinook.o \
        SpringChinook.o \
        Blarney.o \
\
```

(The "\" characters just mean that the statement continues onto the next line.)

Second, add the new .h file to this statement:

```
TroutModelSwarm.o: TroutModelSwarm.[hm] globals.h  \
      FallChinook.h SpringChinook.h Blarney.h HabitatSpace.h \
      FishParams.h DEBUGFLAGS.h
```

Finally, a line must be added to the statements for each species:

```
FallChinook.o : FallChinook.[hm] DEBUGFLAGS.h
SpringChinook.o : SpringChinook.[hm] DEBUGFLAGS.h
Blarney.o : Blarney.[hm] DEBUGFLAGS.h
…
```

These steps add a new species that has exactly the same formulation as the other species (although the new species can have its own parameter values). Differences in formulation among species can be implemented by copying the relevant methods from Trout.h and Trout.m to the species' source code files and revising them.

After these changes are made, the software must be re-compiled to create a new version of the executable file inSALMO.exe.

6.2. Changing Size Classes

The size classes used to summarize model results must be defined in the *inSALMO* source code, but it is easy to modify the code to add or remove size classes. (The number of size classes do not affect simulations at all, only how statistical summaries of results are calculated and reported.)

Most of the changes are made in the file TroutModelSwarm.m. The first is in the following lines, which appear in the method "buildObjects":

```
sizeSymbolList = [List create: modelZone];

Size0to5     = [Symbol create: modelZone setName: "Size0to5"];
[sizeSymbolList addLast: Size0to5];
Size5to8     = [Symbol create: modelZone setName: "Size5to8"];
[sizeSymbolList addLast: Size5to8];
```

```
Size8Plus     = [Symbol create: modelZone setName: "Size8Plus"];
[sizeSymbolList addLast: Size8Plus];
```

These lines define the size classes, and can be edited to change the number and range of classes. For example, this change breaks the size category of 8 cm and higher into two classes (the changes are in bold text):

```
sizeSymbolList = [List create: modelZone];

Size0to5     = [Symbol create: modelZone setName: "Size0to5"];
[sizeSymbolList addLast: Size0to5];
Size5to8     = [Symbol create: modelZone setName: "Size5to8"];
[sizeSymbolList addLast: Size5to8];
Size8to10     = [Symbol create: modelZone setName: "Size8to10"];
[sizeSymbolList addLast: Size8to10];
Size10Plus     = [Symbol create: modelZone setName: "Size10Plus"];
[sizeSymbolList addLast: Size10Plus];
```

The second change must be made in the TroutModelSwarm method "getSizeSymbolForLength", which assigns a size class to a fish, using the fish's length. For the default size classes, this method is:

```
- (id <Symbol>) getSizeSymbolForLength: (double) aLength
{
   int offset = 0;

   if(aLength > 5.0)
   {
      offset = 1;
   }
   if(aLength > 8.0)
   {
      offset = 2;
   }

   return [sizeSymbolList atOffset: offset];
}
```

For the above example in which a 8-10 cm size class is added, the method must be changed to:

```
- (id <Symbol>) getSizeSymbolForLength: (double) aLength
{
   int offset = 0;

   if(aLength > 5.0)
   {
      offset = 1;
   }
   if(aLength > 8.0)
   {
      offset = 2;
```

```
        }
        if(aLength > 10.0)
        {
           offset = 3;
        }

        return [sizeSymbolList atOffset: offset];
     }
```

Finally, the file TroutModelSwarm.h must be edited to declare the new size class symbols. Change this block of code:

```
id <List> sizeSymbolList;
id <Symbol> Size0to5;
id <Symbol> Size5to8;
id <Symbol> Size8Plus;
```

to this:

```
id <List> sizeSymbolList;
id <Symbol> Size0to5;
id <Symbol> Size5to8;
id <Symbol> Size8to10;
id <Symbol> Size10Plus;
```

After these changes are made and the code re-compiled (using `make clean` first), *inSALMO* will automatically use the new classes to report results that are broken out by size class.

7. Graphical Interfaces

When the *inSALMO* software is executed in its default graphics mode, a number of graphical interfaces are provided. Most of these are simply graphs that display results, but the animation window is truly an interactive interface to the model, allowing users to probe deeply to observe and even alter the state of individual habitat cells, fish, and redds. Control panels can be used to start, stop, or step through model runs.

Users just interested in starting up their first model runs only need to know that they must hit the "Start" button on the first control panel *twice* to get execution underway.

It is important to understand that the graphical interfaces are updated only at the end of each daily time step, after all other scheduled actions are complete. Mouse clicks and other input to the interfaces are accepted only at the end of the time step: you cannot stop a simulation part way through a day's schedule, and displayed information reflects the model's state at the end of a time step.

When the model is started in batch mode (via the GUI or by using the command `inSALMO.exe -b`), none of these graphics appear. Instead, simulations start immediately and run until completed (or until something goes wrong, or the user kills the job).

7.1. Main Control Panel

As soon as *inSALMO* is started in graphics mode, the main control panel opens up (Figure 12). This panel essentially operates the Experiment Manager, which then starts the model runs. The Experiment Manager (Section 10) may be set up so only one model run is executed, or so that a multi-run experiment is executed. Users need to know only the following:

Figure 12. Main control panel.

- Model execution is started by hitting the "Start" button once (which creates the Experiment Manager), then a second time (which starts the first model run). After the "Start" button is hit the second time, control of the first model run is passed to the Model Run Controller (described below).

- If the Experiment Manager is set up for a single run, execution can be terminated cleanly by hitting the "Quit" button after the run is finished.

- If the Experiment Manager is set up for multiple runs, the “Start” button must be hit to start each run. After each run finishs, nothing happens until this button is hit again.

- The “Save” button is supposed to tell Swarm to preserve the spatial arrangement of graphical interface windows on the screen and use it next time the code is executed. However, this facility is not reliable and the hidden file it creates can become corrupted and cause frustrating problems. Users are recommended not to use it.

- The “Quit” button stops execution and closes the code.

7.2. Model Run Controller

When a model run has been started from the main control panel, it opens a second control panel labeled “Model Run Controller” (Figure 13). This panel displays the simulation’s current time step (the number of simulation days that have been completed, starting with zero). It also has six buttons that can be used to control execution.

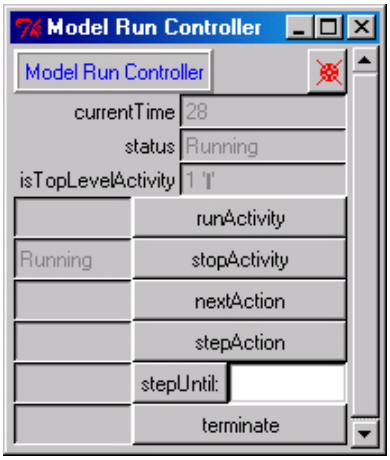


Figure 13. Model Run Controller.

- “runActivity” causes the model to resume execution after it has been paused.

- “stopActivity” causes the model to pause execution (after the current time step has finished).

- “nextAction” causes execution of one more time step, after which execution is again stopped. (Users may have to hit this button a number of times before it executes a full time step; it seems to work well after the “StepUntil” button is used, also.)

- “stepAction” causes execution of one individual action in the model’s schedule (e.g., spawn, move, grow, die).

- “stepUntil” causes execution to continue until it reaches (but does not yet execute) a specified time step. To the right of this button is a window where the time step is input. For example, if the model is stopped at time step 5 and the user wants it to continue for 3 more

days, the number 9 should be entered in the stepUntil input window (by typing “9” then hitting the Enter key), and StepUntil pressed.

- “terminate” causes execution to end the current model run, passing control back to the main control panel. The proper way to shut down execution during a model run is to hit “stopActivity”, then this button, then “Quit” on the main control panel.

7.3. Habitat Probe Displays

A “probe display” window (probe displays are explained in Section 7.5) is opened for each habitat reach. These are small windows (Figure 14) displaying key habitat variables: the current date, flow, temperature, turbidity, etc. The first variable displayed is the reach’s name.

Method probes are in the lower part of the display probes (switchColorRep to unTagAllPolyCells in Figure 14) and have an (initially) blank value window to the left of the method name. A method can be executed by clicking on the button displaying the method’s name. When clicked on, these method probes return a value in the window to the left of the button.

Some method probes execute methods that require a parameter (an input to the method). These probes appear as a button with a window on its right (where the parameter value is entered before clicking the button) and with a window on its left, where the value returned from the method is displayed. The tagCellNumber in Figure 14 is an example: the required parameter is the number of the cell that will be “tagged” by changing its color; this number must be entered in the box to the right of the “stepUntil” button, then the button clicked on.

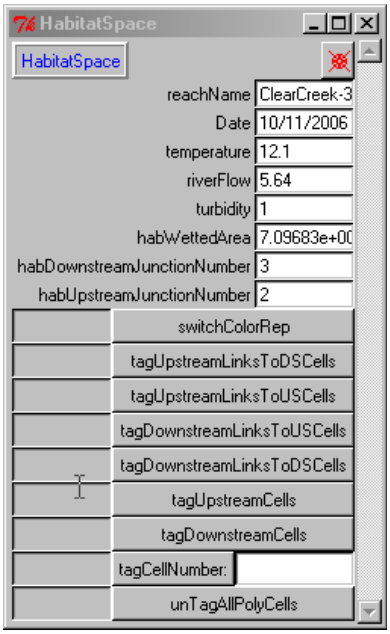


Figure 14. Habitat probe display. One of these windows is opened for each habitat reach.

7.4. Graphs

Several graphs provide summary information on the salmon population. These graphs report the status of all salmon in the model; they are not broken out by habitat reach, species, age, or any other characteristic.

- A line graph (time series plot) of the cumulative number of salmon that have died of each mortality source (Figure 15). The X axis is the number of simulation days. Clicking on any mortality source in the graph's legend highlights the line for that kind of mortality.

- A bar graph showing how many fish are currently alive, by age. (Keep in mind that this version of *inSALMO* assigns an age of 5 to all adults.)

- A line graph showing the number of outmigrants each day.

- Histograms showing the number of redds vs. cell depth and velocities. These histograms show the number of redds in cells in each depth (or velocity) category. (They do not by themselves represent "preference" for depth or velocity because they do not account for how much cell area there is of each depth and velocity.) The number and size of histogram bins is set in the source code file TroutObserverSwarm.m, in the method "buildObjects" (look for the line "velocityHisto = [EZBin createBegin: obsZone]"). By default, both the depth and velocity histograms have 10 bins, each representing 10 cm of depth or 10 cm/s of velocity, with values above 100 reported as outliers.

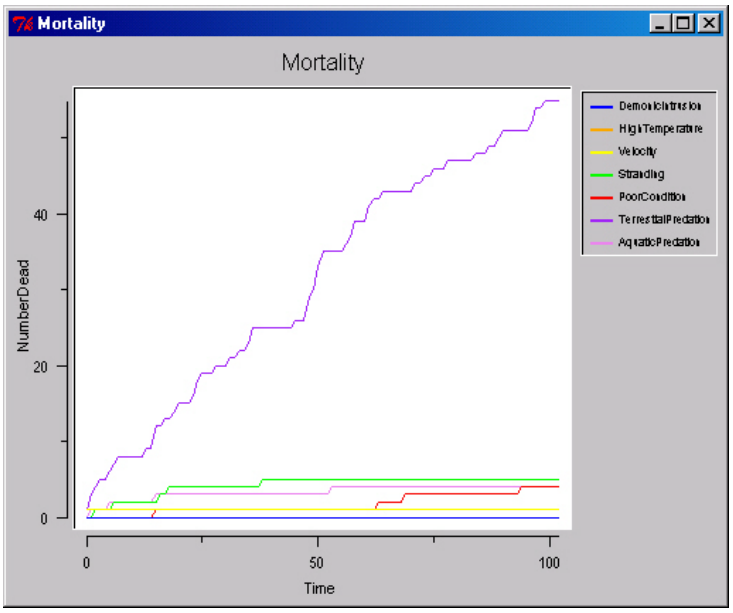


Figure 15. Mortality output graph. "Time" on the X axis is the number of days simulated. The Y axis is the total number of salmon that have died of each of the seven mortality sources.

7.5. Animation Window

The primary interactive interface is the animation window, which shows the habitat cells, fish, and redds as the simulation proceeds (Figure 16). An animation window is opened for each

habitat reach. The window displays the habitat cells as a plan view (looking from the top down) map. The size of the window can be adjusted in both dimensions using parameters in the observer setup file.

The cells are shaded by either depth or velocity, according to the observer setup file parameter *rasterColorVariable*, and whether depth or velocity is used can be changed via the habitat space probe display (Section 7.3). (The shading scheme is defined in the method drawSelfOn in file Cell.m.)

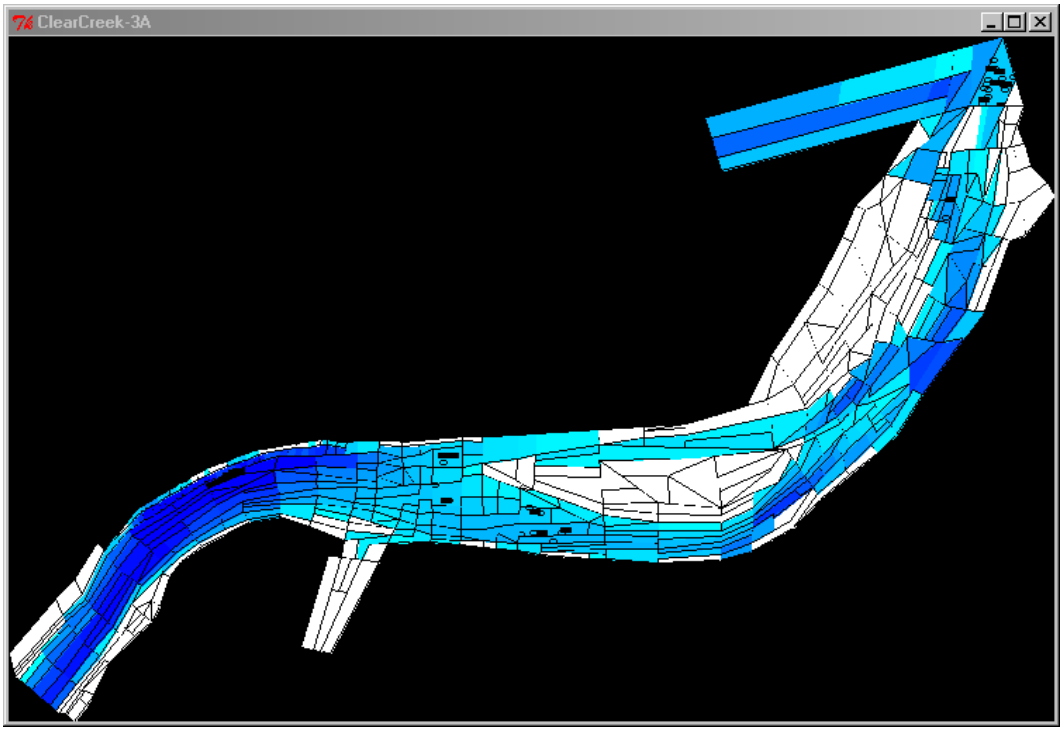


Figure 16. Animation window. One such window is opened for each habitat reach, with the reach name used as the window's title (top left).

Adult fish (age > 0) appear as filled rectangles at a random location in the cell they occupy. Fish colors depend on species and are set in the species setup file. Juvenile fish (age 0) appear as individual pixels (dots) in their cell. Redds appear as an open oval. Redds are also color-coded by species.

7.5.1. Probe displays

The habitat cells, fish, and redds can be "probed" from the animation window, allowing them to be examined in detail and even altered. To open a probe display to a cell, click on it with the left mouse button. A right mouse button click on a cell opens probe displays for all the fish and redds in the cell. It is usually desirable to pausing execution with the model run controller's "stopActivity" button before attempting to open probe displays, but once open they can stay open when execution is resumed, so the user can see how variables change over time. Probe displays should be closed by clicking on the red button near its upper right corner.

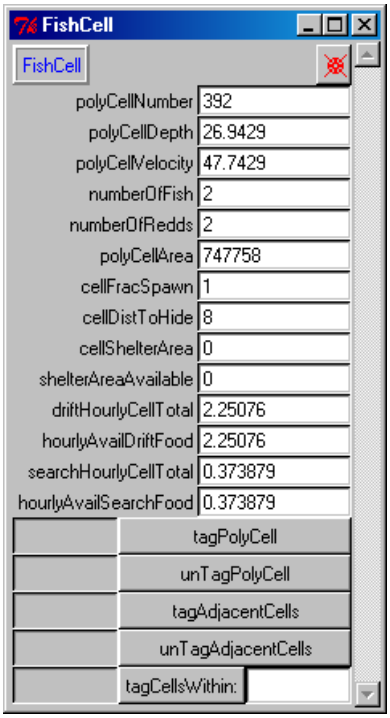


Figure 17. Cell probe display.

Probe displays include two kinds of probes: variable probes allow the user to see and change the value of a particular variable, and method probes allow the user to execute one of the object's methods (a piece of its program that executes some particular function, similar to a subroutine).

Variable probes are in the upper part of the display (polyCellNumber to hourlyAvailSearchFood in Figure 17); variable values show up in little windows to the right of the variable name. The value for a variable can be replaced manually by entering it into the value window over the old value, and then hitting "Enter" (but see the following note about when values are updated).

Note that the animation window is updated <u>after</u> habitat and fish simulations are completed each time step, and <u>before</u> habitat variables (depth, velocity, temperature, flow) are updated for the next time step. Therefore, any time-variable cell variables changed via probes will be re-set and overwritten before the next fish simulations. For example, using the probe to change a cell's depth or velocity will have no effect because these variables are updated with a new daily value as soon as execution is resumed.

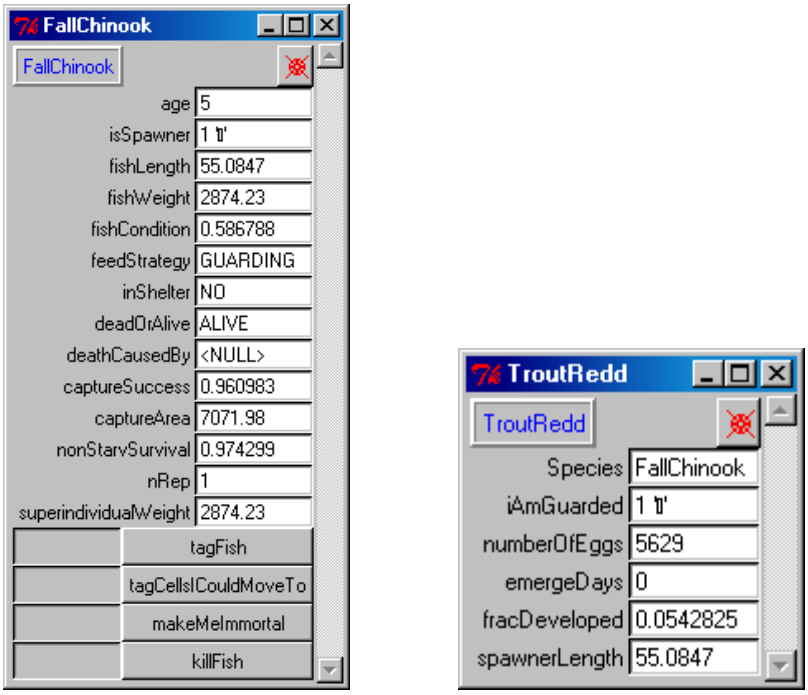


Figure 18. Salmon and redd probe displays.

The default probe display for a salmon includes four method probes that do the following things:

- “tagFish” turns the probed salmon a different color in the animation window, so it can be followed when simulations are resumed. (This color is set in the Model.Setup file.)
- “tagCellsICouldMoveTo” temporarily highlights the cells that are potential movement destinations of the salmon (as defined in the model description). The salmon’s current cell is not highlighted.
- “makeMeImmortal” causes the probed salmon to be exempt from mortality for the rest of the simulation. None of the salmon’s behavior is changed.
- “killFish” causes the salmon to die immediately, with the mortality source being demonic intrusion. (“killFish” trumps “makeMeImmortal”.)

The probe displays that can be opened from the animation window include only a few selected variables and methods. (Variables and methods can be added to or removed from these displays by editing the code in the method “buildProbesIn” or—for salmon—“buildFishProbes”, in the code file “TroutObserverSwarm.m”.) However, a “complete probe display” that shows all of an object’s variables and methods can be opened by right-clicking on the box, at the top left of a probe display window, that states the object’s class (either “Cell”, “Redd”, or the species name). These complete probe displays, like all others, are closed by clicking on the red button at their top right corner.

For a salmon, this complete probe display is empty because the methods and variables for a salmon are in the Trout superclass, not the subclass for each species. However, all the complete probe display windows include a button, at the top right corner of the window, showing two green boxes and an arrow from the lower box to the upper (Figure 19). This button opens a complete probe display of the object's superclass. Therefore, to open a complete probe display for a salmon, right-click on the blue species name box; then, in the newly opened window, click on the green superclass button.

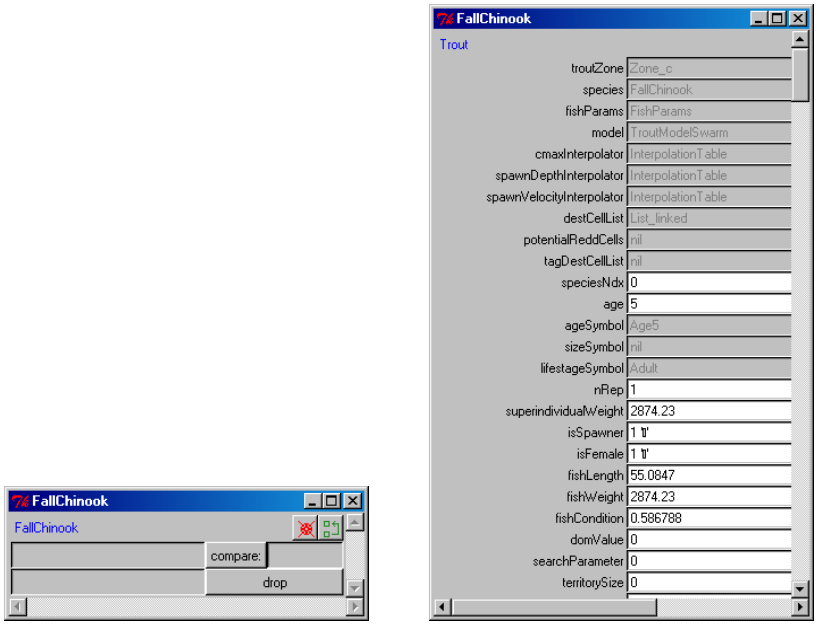


Figure 19. Right-clicking on the species name box in a salmon probe display opens a (empty) complete probe display (left). Clicking on the green button opens a complete probe display for the salmon (right).

7.5.2. Movies of the animation window

It is relatively easy to make a digital "movie" of the animation window (or full screen) during a model run; these movies can be used on web sites, in presentations, or in digital appendices to publications. Movies are made by having *inSALMO* write one graphics output file for each time step, then assembling these files into an animation file.

By default, *inSALMO* produces the graphics output by capturing the animation window for the first reach in the Reach.Setup file (Section 3.3). Alternatively, the entire screen can be captured, as described below. The steps are:

- In the observer setup file, set the parameter *takeRasterPictures* to YES. This will cause *inSALMO* to write, at the end of each daily time step, a new graphics file that contains an image of the animation window. The files are in Portable Network Graphics (.png) format, and have sequential file names ("Model001_Frame001.png", "Model001_Frame002.png", etc.).

- Run the model in graphics mode. To protect the animation output, the animation window now remains on top of any other windows.

- Usually, the model should be stopped after 50-100 time steps to keep from producing too many frames to use in a reasonably sized movie.

- Animation software is used to assemble the frames into a movie. GIF Construction Set Professional, for example, is inexpensive shareware that allows the .png files to rapidly be assembled into movies in either .gif of .avi format.

A simple code change causes *inSALMO* to write out pictures of the entire screen, not just the animation window. In the method writeFrame of source file TroutObserverSwarm.m, simply comment out the statement "[pixID setWidget: raster];" and re-compile the model.

8. File Output

This section describes the main output files that are written for all model runs. These output files provide the kind of detailed summary statistics that are usually most useful for understanding what happened during a simulation.

8.1. General Information on Output Files

This section briefly describes characteristics common to the main output files. All output files are written in CSV format so they are very easily opened in spreadsheet software. (See Section 2.1 concerning CSV files.)

The three fish output files are generated using the EcoSwarm BreakoutReporter, a generic output file-writing tool. The BreakoutReporter makes it easy to modify the software to get more, less, or different detail in the output files. By changing a few simple statements in the code, users can (1) output additional variables, (2) output different summary statistics (minimum, mean, maximum, count, sum, standard deviation, variance) for a variable, and (3) change the fish characteristics (e.g., species, age class, habitat reach) by which results are broken out or the order in which results are broken out (e.g., species first, then age; vs. age first, then species). The code for these output files is in the method createBreakoutReporters in the class TroutModelSwarm. (For example, code to also output the minimum and maximum length and weight of fish is currently written, but commented out, in createBreakoutReporters.) Documentation for the BreakoutReporter is at www.humboldt.edu/ecomodel/software.htm.

Starting with version 1.0 of *inSALMO*, the BreakoutReport produces output designed for use with Excel's "PivotTable" feature and statistical software that can summarize data identified by categorical variables. Instead of producing separate columns of output for each category of results (each combination of species, site, size class, etc.), the output files now report all results in one column, with separate columns indicating the category (species, site, etc.) corresponding to the output line.

Output files report the scenario and replicate numbers generated by the Experiment Manager (Section 10). These numbers are needed to separate results from multiple model runs generated by the Experiment Manager.

If a model run stops before it is finished (because an error occurs or because it is killed by the user), the output files will persist and provide results up to when the simulation stopped.

Users should pay attention to the *appendFiles* parameter and how it is set in Model.Setup and Experiment.Setup. This parameter controls whether output files are overwritten vs. appended each time a model run (except the separate model runs within an experiment generated by the Experiment Manager) starts.

The parameter *fileOutputFrequency* in Model.Setup controls how often output is written to the two fish output files. Using a higher value of *fileOutputFrequency* can reduce execution time (because generating the fish outputs using BreakoutReport requires quite a few computations) and reduce the volume of output to be stored and analyzed.

8.2. Output File Descriptions

The main output files are described here. New users can most easily view these files by opening them in spreadsheet software.

Live fish output file. This file reports a time series of summary statistics on the live salmon population. The file name is provided by user, via the parameter *fishOutputFile* in the model setup file (Section 3.4). By default, this file provides the abundance, mean weight, and mean length of the salmon population, broken out by habitat reach, salmon species, and age class.

Fish mortality output file. This file reports mortality statistics: the number of salmon that have died via each mortality source. Results are broken out by habitat reach, species, and age class. On each output date, a line is written to this file reporting the number of salmon that have died of each type of mortality since the previous output date.

Outmigrant output file. “Outmigrants” are juvenile salmon that have decided to move downstream beyond the downstream-most reach in the model. File output on outmigrant numbers, size, and timing are essential for comparison of model results to field data (e.g., screw trap data on outmigration). Outmigrant results are also essential for evaluating simulated river management alternatives because the number of large outmigrants is the primary measure of the salmon population’s success in this version of *inSALMO*.

The outmigrant file is called “Outmigrants_Out.csv” and is updated with the same frequency (e.g., every 10 simulated days) as the main live fish output file. It is updated using the “replacement” mode, from a list containing only the fish that have outmigrated since the last update. Therefore, it reports statistics on the recent outmigrants, not cumulative statistics on all outmigrants over the simulation. The output file reports the date, then the number of outmigrants broken out by two variables.

The first breakout variable is the name of the reach that the outmigrant was spawned in—its “natal reach”. The second breakout variable is a size category assigned when fish outmigrate. There are only three of these categories: “0-4”, “>4-8”, and “>8”, with the names referring to lengths in cm. For example, fish with length less than or equal to 4 cm would be assigned the category “0-4”. (Outmigrants in the first category can be treated as the “moribund fry” often observed in large numbers; those in the third category can be treated as likely successful smolts.) (Changing these size categories is discussed at Section 6.2.)

Redd output file. This file provides one line of results for each redd, written on the day when the redd becomes empty (all eggs have died or turned into new salmon). The file name is specified by the user, via the parameter *reddOutputFile* in the model setup file. Results reported for each redd include the spawner's length, weight, and age; the dates the redd was created and emptied; the redd's location (reach and cell); the initial number of eggs; and the number of eggs dying of each redd mortality source and the number emerging as new salmon. The output also includes a "ReddID", which is simply a unique alphanumeric code for each redd.

Habitat output files. These files output the values of the time-series habitat variables. One file is written for each habitat reach; its name is the reach's name with "Habitat.out" appended to it (e.g., MainstemReachHabitat.out). Each line reports the date and the daily flow, temperature, and turbidity values (which were input via the files described in Section 5.5).

These habitat output files are actually optional (Section 9) and can be suppressed by commenting out the statement `#define HABITAT_REPORT_ON` in the code file HabitatManager.h.

Depth use histogram output. This file is simply a file version of the depth histogram described in Section 7.4. Its file name is always ReddDepthHisto.out. The file (generated by Swarm's histogram tool) contains no column headings, and includes one line for each simulation day. Each line includes values, separated by tab characters, equal to the number of redds in each histogram bin. Note that outliers (redds in depths greater than the histogram's upper bound) are ignored in this file, so the total number of redds on a line of output may be less than the total number alive. (More useful output for analysis of habitat availability and habitat use by model salmon is provided by the optional CellFishInfo.rpt output file described in Section 9.)

This file is not generated when *inSALMO* is executed in batch mode, and is always overwritten each model run (including each replicate of each scenario if the Experiment Manager is used).

Velocity use histogram output. This is file output of the velocity histogram described in Section 7.4, and always named ReddVelocityHisto.out. The file format and characteristics are the same as those of the depth histogram.

9. Optional Output Files for Testing and Specialized Studies

The *inSALMO* software includes a number of optional output files that are normally not written, but can be turned on when more detailed output is desired. These "reports" are often useful for testing changes in the software or parameters, and for supporting more detailed analysis of salmon behavior. However, many of these optional output files can be extremely large and writing them can make *inSALMO* much slower to execute.

The optional output files are turned on by making very simple edits in several of the code files. Normally, the statement that activates each of these files is commented out by placing two slashes in front of it. These statements appear in several of interface (.h) source code files, and include the word `#define` and the report name. For example, a report providing details on redd survival can be turned on by editing the file "TroutModelSwarm.h" and changing this line:

```
//#define REDD_SURV_REPORT
```

to:

```
#define REDD_SURV_REPORT
```

Then the source code must be recompiled. When changes are made to any interface files, it is best to recompile using `make clean` first, then `make`.

Table 4 provides a complete list of the optional output files. The first column tells where in the source code—which statement in which .h file—the file is activated. The second column provides the name of the output file. Some of the optional reports produce a separate output file for each habitat reach, in which case the file name starts with the reach name. The third column describes the information provided in the file.

With one exception, these optional output files are not controlled by the Model.Setup parameter *appendFiles*; instead, they are always overwritten at the start of each model run. Even for multiple model runs generated by the Experiment Manager (Section 10), these files will report results only from the last model run.

The exception is the CellFishInfo.rpt output (the last line of Table 4). This output is controlled by *appendFiles*, and is also designed so that results from multiple runs generated by the Experiment Manager are automatically appended.

Table 4. Optional output files.

Source code file and activating statement	Report file name	Description of report
Cell.h `#define FOODAVAILREPORT`	FoodAvailability.rpt	Produces one line each time a fish moves into a cell. Reports the cell's hourly food production rate, the hourly rate of food availability (unused by fish already in the cell), and the amount consumed by the fish. Results are separate for drift and search food types.
HabitatManager.h `#define DEPTH_REPORT_ON`	(reachname)Cell_Flow_Depth_Test.rpt	Produces one file for each habitat reach. One line is written at the start of each day, reporting the flow, and depth in each cell.
HabitatManager.h `#define VELOCITY_REPORT_ON`	(reachname)Cell_Flow_Velocity_Test.rpt	Like the depth report, but cell velocity is reported.
HabitatManager.h `#define DEPTH_VEL_RPT`	(reachname)CellDepthAreaVelocity.rpt	Produces one file for each reach. On each day, one line is written for each habitat cell with depth above zero; cell area, depth, and velocity are output.
Trout.h `#define MOVE_REPORT_ON`	MoveTest.rpt	Writes one line when each salmon selects the habitat cell it will occupy, each day. Reports the data (about the selected cell and the salmon) used to select this best cell. Also reports intermediate calculations in the salmon's movement decision (e.g., its respiration rate, net energy intake). Note: The output is first produced when the fish are initialized on the first simulated day, when fish are not yet sorted in size order.
Trout.h `#define READY_TO_SPAWN_RPT`	Ready_To_Spawn.rpt	Produces one line per day for each female salmon. Reports whether the female decided it was ready to spawn and the variables used to make the

		decision.
Trout.h `#define SPAWN_CELL_RPT`	Spawn_Cell.rpt	Produces one line of output each time a female salmon spawns and builds a redd. For each potential spawning cell, reports the cell depth, velocity, and fraction spawning gravel; and the calculated variables used to rate spawning cells: depth and velocity suitability, overall spawning quality.
TroutModelSwarm.h `#define REDD_MORTALITY_REPORT`	Redd_Mortality.rpt	Provides a daily egg mortality report for each redd. When each redd is empty, it writes its report to the end of this file. The report includes the ReddID code, the initial number of eggs, and (on separate lines) the number of eggs dying of each mortality source on each day of the redd's existence.
TroutModelSwarm.h `#define REDD_SURV_REPORT`	ReddSurvivalTest.rpt	Produces a report on egg survival for a redd on the day when the redd has no more live eggs (due to mortality and emergence). Reports the redd's species, location (reach and cell numbers), and initial number of eggs. Then, for each day of the redd's existence, a line reports the temperature, flow, depth, and number of eggs dying of each mortality source.
TroutModelSwarm.h `#define PRINT_CELL_FISH_REPORT`	(reachname)CellFishInfo.rpt	Produces one file for each reach. On each day, writes one line for each habitat cell. Reports cell area, depth, velocity, distance to hiding cover, fraction with velocity shelter, and statistics on the fish in the cell*.

*This file is produced using the EcoSwarm BreakoutReporter, and is very similar to the live fish output file (Section 8.1). The fish statistics provided in this file normally include the number of fish in the cell, broken out by species and age class. However, these statistics are easily

modified as described in Section 8.1, by modifying the BreakoutReporter code in method buildCellFishInfoReporter in file HabitatSpace.m.

10. Experiment Manager

All users of *inSALMO* need to at least be aware of the Experiment Manager because it has powerful features that are often extremely useful, but it can also act like a nightmarish bug if ignored. These features overwrite values provided in setup and parameter files, causing the model's behavior to be inexplicable if one forgets to pay attention to the Experiment Manager. But any serious user of *inSALMO* will quickly learn to depend on the Experiment Manager to conduct simulation experiments quickly, easily, and reliably.

10.1. What the Experiment Manager Does

The purpose of the Experiment Manager is to allow users to set up and execute simulation experiments that use multiple model runs with different inputs, without having to modify either the software or the parameter files. With a few simple statements in its setup file, a complex experiment can be set up to run automatically. The experiments can include both scenarios and replicates. A scenario is a single set of inputs and parameter values; different scenarios are defined by specifying which inputs or parameters differ among them. Replicates are repeated runs of the same scenario, with only the pseudorandom numbers (which primarily affect mortality) differing among replicates.

The Experiment Manager has two functions. The first is to set up and execute the number of simulations specified in its setup file. For example, the user may set up the Experiment Manager to define 10 scenarios (perhaps five different values for some parameter, for each of two flow input files) and three replicates. The Experiment Manager would determine that 30 model runs are needed and start one run after another. The second function is to modify the parameters for each of the model runs to implement the scenarios. This function occurs at the start of a model run: the Experiment Manager stops the model after parameter values have been read in, then overwrites the value of any parameters that are specified in its setup file. For replicates, the Experiment Manager simply changes the random number seed (the seed value provided in Model.Setup is multiplied by the replicate number). The Experiment Manager then re-starts the model and has no more effect until the next run starts. (To be precise: the TroutModelSwarm method instantiateObjects is executed; then the Experiment Manager is executed and modifies parameter values; then the TroutModelSwarm method buildObjects is executed and the simulation proceeds.)

The Experiment Manager's scenarios almost always involve modifying the value of variables that are in setup or parameter files. The Experiment Manager is generally not useful for directly modifying variables that are in input files (flows, temperatures, cell characteristics, etc.); however, it works very nicely to define such scenarios by creating different input files and using the Experiment Manager to control which input file is used.

The following sections describe the setup file that controls the Experiment Manager, and provide many examples that can be followed to set up experiments.

10.2. General Procedure for Setting Up Experiments

The general procedure for using the Experiment Manager is to (1) specify the scenarios and replicates in the Experiment.Setup file, (2) run *inSALMO* in batch (non-graphics) mode, and (3)

examine and analyze the results. Results are usually written to one file, with output labeled by scenario and replicate number. It is always good to archive a copy of the Experiment.Setup file with the output files from an experiment, to document exactly what scenarios were executed.

10.2.1. Experiment.Setup format

The Experiment.Setup file controls the Experiment Manager. This file is always read by *inSALMO*, so it must be configured correctly even if the model is to be run with no automated experiments. This format of this setup file is described here and illustrated via several examples in Section 10.3 (see also Figure 20). Experiment.Setup contains:

- Comments, which can be included anywhere as lines that start with the character "#".
- Three header lines, not used by the computer. These can be up to 200 characters long.
- A blank line
- Two lines on which (a) the number of scenarios and (b) the number of replicates for each scenario are specified. These values must be at least one. There are no built-in limits to how many scenarios or replicates can be used.
- A blank line
- Two lines that provide the variable name and class to which the code sends the current scenario count, during model execution. These lines should not be changed.
- A blank line
- Two lines that provide the variable name and class to which the code sends the current replicate count. These lines should also not be changed.
- A blank line

Following these initial blocks of text, the file contains zero or more additional blocks, which each specify a model parameter to be varied among scenarios and the value it has for each scenario. (At least one such block must be specified if the number of scenarios is greater than one.) There is no limit to how many of these blocks can be specified, or how many parameters can be controlled by the Experiment Manager. These blocks contain the following lines; blocks are separated by a blank line.

- The word `ClassName` followed by the name of the class in which the parameter value is defined. "Class" refers to the object-oriented software structure in which each object in the model is an instance of a particular class. (Class, instance, and variable names are further explained in Section 10.2.2.)
- The word `InstanceName` followed by the name of the instance of the class for which the parameter is to be varied. If the word `NONE` is provided for InstanceName, then the parameter is varied for all instances of the class.

- The word `ParamName` followed by the name of the parameter to be varied.
- The word `ValueType` followed by the kind of value that the parameter contains. The value type must be one of the types defined in Section 10.2.3.
- The word `Value` followed by the parameter's value for the first scenario. This line is repeated for each scenario: there must be one value provided for each scenario, even if the value is the same for some scenarios.

```
Experiment setup file -
Created Feb 21 2005
For demo example

numberOfScenarios    3
numberOfReplicates   5

sendScenarioCountToParam: scenario
inClass:                  TroutModelSwarm

sendReplicateCountToParam: replicate
inClass:                  TroutModelSwarm

ClassName      FishParams
InstanceName   FallChinook
ParamName      fishFitnessHorizon
ValueType      double
Value          60.0
Value          90.0
Value          120.0
```

Figure 20. Example Experiment.Setup file.

10.2.2. Class and instance names for typical experiments

The Experiment Manager is highly flexible, allowing users access to any instance variable of any class in *inSALMO*. However, using it successfully for unusual experiments requires detailed knowledge of the software; some inputs cannot be usefully manipulated because they have already been used before the Experiment Manager executes (e.g., the hydraulics data files) or because their values are overwritten after the Experiment Manager executes (e.g., reach flow; cell depth and velocity). Table 5 describes the usual applications of the Experiment Manager that can be made with confidence.

Table 5. Class and instance names for parameters commonly used in the Experiment Manager.

Parameters	ClassName	InstanceName
Model setup parameters (any parameters in Model.Setup)	TroutModelSwarm	NONE
File names for cell data, flow, temperature, and turbidity input (parameters *flowFile*, *temperatureFile*, *turbidityFile* in the reach setup file)	HabitatSpace	The reach name specified in Reach.Setup (or NONE if only one reach is simulated)
Habitat parameters (any parameter in a habitat parameter file)	HabitatSpace	The reach name specified in Reach.Setup (or NONE if the Experiment Manager is to alter the parameter for all habitat reaches)
Salmon and redd parameters (any parameter in a salmon parameter file)	FishParams *	The species name specified in Species.Setup (or NONE if the Experiment Manager is to alter the parameter for all species)

*Salmon parameter values are not stored in the model salmon themselves, but in a separate "FishParams" object for each species.

Input that <u>cannot</u> be manipulated by the Experiment Manager include parameters in the Observer.Setup and Species.Setup files and the hydraulics data file names in Reach.Setup.

10.2.3. Valid parameter value types

The Experiment.Setup file must provide a "ValueType" for each parameter to be manipulated. Valid value types are defined in Table 6. The value type depends on the parameter itself. If users are not sure what type a parameter is, they should find the parameter in the header (.h) file for the parameter's class. For example, the parameter controlled by the setup file in Figure 20 is *fishFitnessHorizon*, the number of days over which a fish evaluates its habitat selection decision. This parameter could be either an integer or floating point (real) number. Searching the file FishParams.h for *fishFitnessHorizon* finds the declaration "const double fishFitnessHorizon", indicating that this parameter is a double-precision floating point number, so ValueType should be "double".

Table 6. Value types for the Experiment.Setup file.

Value type*	Definition
`BOOL`	A boolean variable, with a value of either `YES` or `NO` (all upper case)**
`date`	A date variable in MM/DD/YYYY format
`day`	A day-of-the-year variable in MM/DD format
`double`	A double-precision floating point variable (any number that is not an integer)
`filename`	The name of an input file (a character string)
`int`	An integer variable

*The "ValueType" field in an Experiment.Setup file must exactly match one of the values in this column, including upper/lower case.

**Note that boolean variables must have values of `0` or `1` in other setup files, but values of either `YES` or `NO` in Experiment.Setup.

10.2.4. Using instance names

The InstanceName field in Experiment.Setup allows the user to manipulate parameters for one instance of a class: for one of several salmon species, or for one of several habitat reaches (see Table 5). InstanceName is set to `NONE` if there is only one instance per class, or if the same parameter change is to be made for all instances.

Separate parameter blocks can be used in the Experiment.Setup file to change the same parameter different ways for different species or habitat reaches. The following examples illustrate how the Experiment Manager uses instance names. The examples assume a version of *inSALMO* with three salmon species named LateFallChinook, FallChinook, and SpringChinook; and that only one scenario is generated.

Example 1 changes the value of parameter *fishEnergyDensity* to 4555, but only for LateFallChinook; other species retain the value in their parameter files:

```
ClassName      FishParams
InstanceName  LateFallChinook
ParamName    fishEnergyDensity
ValueType      double
Value              4555
```

Example 2 changes the value of *fishEnergyDensity* for all species:

```
ClassName      FishParams
InstanceName  NONE
ParamName    fishEnergyDensity
ValueType      double
Value              4555
```

Example 3 changes the value of *fishEnergyDensity* to 4555 for LateFallChinook salmon and to 5000 for SpringChinook; FallChinook are unaffected:

```
ClassName      FishParams
InstanceName   LateFallChinook
ParamName      fishEnergyDensity
ValueType      double
Value          4555

ClassName      FishParams
InstanceName   SpringChinook
ParamName      fishEnergyDensity
ValueType      double
Value          5000
```

Example 4 changes *fishEnergyDensity* to 4555 for LateFallChinook and to 5000 for all other species. The order in which these two blocks appear does not matter—changing parameter values for a specific instance always overrides a general change (via `InstanceName NONE`) to all instances.

```
ClassName      FishParams
InstanceName  LateFallChinook
ParamName    fishEnergyDensity
ValueType      double
Value              4555

ClassName      FishParams
InstanceName  NONE
ParamName    fishEnergyDensity
ValueType      double
Value              5000
```

Example 5 changes *fishEnergyDensity* to 5000. It does not cause an error even though *fishEnergyDensity* is included twice—if the same parameter is included several times in Experiment.Setup, in the same way, the last value is used.

```
ClassName      FishParams
InstanceName  LateFallChinook
ParamName    fishEnergyDensity
ValueType      double
Value              4555

ClassName      FishParams
```

```
InstanceName  LateFallChinook
ParamName    fishEnergyDensity
ValueType      double
Value              5000
```

10.2.5. Controlling where output goes

Normally the Experiment Manager is set up so output from all scenarios and replicates are sent to the same output files, which are automatically appended for each model run (even if the parameter *appendFiles* is set to `0` in Model.Setup). The standard output files include the scenario and replicate number for all output.

An alternative is to include unique names for output files for each scenario in the Experiment.Setup file. This would write results from each scenario to a different output file (but multiple replicates would still be in the same file). (This capability has not been tested and no examples are provided.)

10.2.6. Checking the Experiment Manager

The Experiment Manager includes several kinds of error checking; execution stops with an error statement if:

- The number of scenarios or replicates is set to zero;
- A ValueType field has an invalid value, or its value does not match that of the parameter;
- A ClassName or InstanceName field has an invalid value;
- The named parameter does not exist in the specified class; or
- The number of values provided for a parameter is not exactly equal to the number of scenarios.

There are several ways to verify that the Experiment Manager produced the intended parameter values. Manipulations of input data (e.g., the flow or temperature input file) can be checked by examining the habitat output files (Section 8.2).

Habitat parameters can be checked by running *inSALMO* in graphics mode and using the HabitatSpace probe display (Section 7.5.1). (Remember that multiple scenarios can be run in graphics mode by clicking on “Start” on the main control panel after each run finishes; and that all habitat variables can be viewed by right-clicking on the box labeled HabitatSpace in the top left corner of the probe display; Figure 14.)

Salmon parameters can be checked by turning on an optional output that prints out salmon parameter values at the start of each model run (after they have been manipulated by the Experiment Manager). A separate output file is created for each species; these are named SpeciesXParamCheck.out, where X is the name of the salmon species. There are two ways to turn this optional output on; one is to add this line to Model.Setup:

```
printFishParams   1
```

The second way is by controlling this parameter via the Experiment.Setup file, using a block such as this:

```
ClassName     TroutModelSwarm
InstanceName  NONE
ParamName     printFishParams
ValueType     BOOL
Value         NO
Value         YES
```

This example prints out fish parameters only at the start of the second scenario. Note that this output is overwritten each model run, so it only reflects the last scenario started.

10.3. Example Experiment.Setup Files

The Experiment Manager is fairly complicated, so a number of examples are provided here. Most users should be able to design the experiments they need by modifying these Experiment.Setup files.

10.3.1. No experiment

When users want to run a single model run, with no parameters altered by the Experiment Manager, the following Experiment.Setup file can be used.

```
Experiment setup file -
Created Feb 21 2005
Example for de-activated Experiment Manager

numberOfScenarios    1
numberOfReplicates   1

sendScenarioCountToParam: scenario
inClass:                  TroutModelSwarm

sendReplicateCountToParam: replicate
inClass:                  TroutModelSwarm
```

10.3.2. Replicate simulations

Users often simply want to run one scenario several times as replicates. Replication is useful just to understand how stochastic model results are. This Experiment.Setup will run the selected number of replicates (five, in this example) and append results from each model run to the output files.

```
Experiment setup file -
Created Feb 21 2005
Example for replication of one scenario
```

```
numberOfScenarios    1
numberOfReplicates   5

sendScenarioCountToParam: scenario
inClass:                  TroutModelSwarm

sendReplicateCountToParam: replicate
inClass:                  TroutModelSwarm
```

10.3.3. Parameter sweep for salmon

One of the most common kinds of experiment is a “parameter sweep”: an experiment in which one parameter is varied over a wide range. The example uses the salmon parameter *mortFishTerrPredMin*, and applies the sweep to all salmon species in the model. Therefore, this example could be used how salmon populations respond to increasing levels of terrestrial predation risk (decreasing survival probability).

```
Experiment setup file -
Created Feb 21 2005
Example salmon parameter sweep

numberOfScenarios    6
numberOfReplicates   1

sendScenarioCountToParam: scenario
inClass:                  TroutModelSwarm

sendReplicateCountToParam: replicate
inClass:                  TroutModelSwarm

ClassName      FishParams
InstanceName   NONE
ParamName      mortFishTerrPredMin
ValueType      double
Value          0.950
Value          0.961
Value          0.972
Value          0.983
Value          0.994
Value          1.0
```

10.3.4. Habitat parameter sweep

This parameter sweep varies a habitat variable: the drift food concentration in one habitat reach (“MainstemUpperReach”). This example illustrates an experiment to see how salmon populations respond to food availability in one of several reaches. This example also illustrates that parameter values can be in scientific notation.

```
Experiment setup file -
Created Feb 21 2005
Example habitat parameter sweep
```

```
numberOfScenarios    6
numberOfReplicates   1

sendScenarioCountToParam: scenario
inClass:                  TroutModelSwarm

sendReplicateCountToParam: replicate
inClass:                  TroutModelSwarm

ClassName      HabitatSpace
InstanceName   UpperMainstem
ParamName      habDriftConc
ValueType      double
Value          5.0E-11
Value          7.0E-11
Value          9.0E-11
Value          1.10E-10
Value          1.30E-10
Value          1.50E-10
```

10.3.5. Multiple parameter sweep

This experiment explores interactions among variables: what happens if two parameters are varied such that all combinations are simulated? This example varies two parameters (for fish vs. terrestrial predation risk), with three values of each; but the same approach can be used with more parameters and more values.

```
Experiment setup file -
Created Feb 21 2005
Example multiple parameter sweep

numberOfScenarios    9
numberOfReplicates   1

sendScenarioCountToParam: scenario
inClass:                  TroutModelSwarm

sendReplicateCountToParam: replicate
inClass:                  TroutModelSwarm

ClassName      FishParams
InstanceName   NONE
ParamName      mortFishAqPredMin
ValueType      double
Value          0.95
Value          0.95
Value          0.95
Value          0.97
Value          0.97
Value          0.97
Value          0.99
Value          0.99
Value          0.99
```

```
ClassName      FishParams
InstanceName   NONE
ParamName      mortFishTerrPredMin
ValueType      double
Value          0.95
Value          0.97
Value          0.99
Value          0.95
Value          0.97
Value          0.99
Value          0.95
Value          0.97
Value          0.99
```

10.3.6. Alternative daily input files

A primary application of *inSALMO* is to compare alternative *inSALMO* flow and temperature scenarios. This example illustrates how three different streamflow regimes can be contrasted, by creating three alternative flow input files and using the Experiment Manager to generate replicate simulations of each. (The flow input files could, for example, be generated by a reservoir model simulating alternative reservoir operating rules.) This example applies the same flow input to all habitat reaches, including when only one reach is simulated.

Note that while the input files names for flow, temperature, and turbidity are provided in the Reach.Setup file (Section 3.3), they are passed to the habitat reach objects (class HabitatSpace) before the Experiment Manager is activated. Therefore, the Experiment Manager accesses these file names in HabitatSpace. The parameter names for these input files are the same in HabitatSpace as they are in Reach.Setup (*flowFile*, *temperatureFile*, *turbidityFile*).

```
Experiment setup file - Comparison of alternative flow scenarios
Created Feb 21 2005
Three flow scenarios, five replicates of each

numberOfScenarios    3
numberOfReplicates   5

sendScenarioCountToParam: scenario
inClass:                  TroutModelSwarm

sendReplicateCountToParam: replicate
inClass:                  TroutModelSwarm

ClassName      HabitatSpace
InstanceName   NONE
ParamName      flowFile
ValueType      filename
Value          FlowScenario1.Data
Value          FlowScenario2.Data
Value          FlowScenario3.Data
```

10.3.7. Multiple simulation years

This example shows how to run the model one year at a time, but with separate runs for several different years. The Model.Setup file includes the parameters controlling when the model starts and stops. In this example, five replicates of three year scenarios are run.

This example is also easily modified to manipulate other parameters in Model.Setup.

```
Experiment setup file - for alternative years
Created Feb 21 2005
Example for changing model setup parameters

numberOfScenarios    3
numberOfReplicates   5

sendScenarioCountToParam: scenario
inClass:                  TroutModelSwarm

sendReplicateCountToParam: replicate
inClass:                  TroutModelSwarm

ClassName    TroutModelSwarm
InstanceName   NONE
ParamName    runStartDate
ValueType    date
Value        10/1/1999
Value        10/1/2000
Value        10/1/2001

ClassName    TroutModelSwarm
InstanceName   NONE
ParamName    runEndDate
ValueType    date
Value        9/30/2000
Value        9/30/2001
Value        9/30/2002
```

10.3.8. Alternative cell data files

Different cell data files might be used to simulate different availabilities of hiding or feeding cover or spawning gravel. The cell data file is a variable of the HabitatSpace class, and the instance name is the name of the habitat reach as defined in Reach.Setup.

```
Experiment setup file - for alternative cell data files
Created Apr 20, 2005
Example

numberOfScenarios    2
numberOfReplicates   5

sendScenarioCountToParam: scenario
inClass:                  TroutModelSwarm
```

```
sendReplicateCountToParam: replicate
inClass:                   TroutModelSwarm

ClassName      HabitatSpace
InstanceName   MiddleReachBearCreek
ParamName      cellDataFile
ValueType      filename
Value          HiCoverCell.Data
Value         LoCoverCell.Data
```

11. Trouble-shooting Guide

Table 7 lists the symptoms and solutions for common problems encountered in trying to install and run *inSALMO*. Additional help for Swarm models in general is available from the Frequently Asked Questions resources on the Swarm web site, www.swarm.org.

Table 7. Trouble-shooting guide.

Symptom	**Potential Cause and Solutions**

When I try to compile the model, the compiler says something like “Makefile: ... Makefile.appl: No such file or directory. make: *** No rule to make target ...”	You might just not be in the right directory, where the code files and Makefile (explained below) are.
	
	The environment variable “SWARMHOME” is not set correctly, most likely because you (1) forgot to set this environment variable, or (2) set it to the wrong directory, perhaps because Swarm was installed to a non-standard directory. (If you are working in Windows with the MinGW release of Swarm, you should have a Windows environment variable set to `/c/swarm`.
	
	If you cannot figure out how to set SWARMHOME correctly, do this: (1) Open the makefile in an editor. In the directory of source code for *inSALMO* will be a file named “Makefile”, which contains the directions MinGW uses to compile the model. This is a plain text (ASCII) file that you can edit using Notepad or Wordpad. At the top of the makefile you should see several lines similar to:
	
	`ifeq ($(SWARMHOME),)`
	`SWARMHOME=/usr`
	`endif`
	
	(2) Comment those lines out by putting a “#” character in front of them, and add a new line saying exactly where SWARMHOME is. The result should be:
	
	`#ifeq ($(SWARMHOME),)`
	`#SWARMHOME=/usr`
	`#endif`
	`SWARMHOME=/c/swarm`
	
	where the last line points to `C:\swarm`, where the Swarm libraries must be located. Make sure you use forward slashes “/”. See

	www.swarm.org/index.php/Swarm_and_MinGW for more information on installing Swarm. See the following entry for a problem likely at this point.
I edited the Makefile (or a code file or input file) using a Windows editor (e.g., Notepad, Wordpad, Word), and saved the change; but *inSALMO* (or MinGW) ignores the change.	Check whether the editor saved the changes in a new file called “Makefile.txt” instead of in the original file “Makefile”. These editors sometimes insist that all plain-text files should end in “.txt”. You can overcome this insistence by putting the file name in quotation marks when telling the editor where to save the file.
When I open output files in Windows Notepad, the lines are all run together.	You opened a file created in Unix format. (This should not happen to Windows-only users.) Use the Linux “unix2dos” conversion facility to convert individual files to Windows format (in Linux, or MinGW, type “unix2dos filename”); or open the files using Wordpad, Word, or Excel, which can handle Unix format.
inSALMO says it cannot find an input file that really is there.	Make sure the file name is completely correct, <u>including upper vs. lower case</u>. File names cannot include blanks. If you are in Linux, was the file created in Windows? If so, you must use the "dos2unix" utility to convert it to Unix file format; otherwise, *inSALMO* will not be able to read it. (But if you accidentally use dos2unix on the executable file `inSALMO.exe`, it will be ruined and you will have to recompile it.)
The model will not read a text parameter such as a file name.	See the previous line concerning file names. Make sure there are no blanks after the parameter value. Make sure the setup file containing the parameter is in the proper Unix or DOS file format.
When I try to start the model by typing `inSALMO.exe` I only get the error “bash: inSALMO.exe: command not found”.	Type `./inSALMO.exe` to start the model.
I changed a parameter value in the salmon (or habitat) parameter file, but the change has no effect.	Check the Experiment.Setup file to see if the parameter value is being controlled by the Experiment Manager.

I edited an input file in Excel (or another spreadsheet program) to make a small change, and now the model crashes (with or without giving a useful error statement). (Similar problems can result from software other than Excel.)	Open the file you edited with a **text editor** such as WordPad (**not** Excel) and inspect it carefully for problems that Excel caused. Are there double quotes are text or other values? If so, remove them. Did small numbers (e.g., 0.0001) get rounded off (e.g., to 0.0)? Did any dates or large numbers get saved as "`######`"? Did dates get changed in format from the required MM/DD/YYYY (e.g., from `10/01/2009` to `10/01/09`)?
After I edit an input file in a spreadsheet, the model runs for a while then stops; the error statement says that something is wrong with dates.	See the end of Section 5.5.
When I start *inSALMO* up, it stops while initializing a model run with an error statement saying that a date was improperly formatted. But all the dates in my input files seem to be correct.	There have been mysterious problems with *inSALMO*'s date/time management software, on some operating systems some of the time. If using Windows, first make sure all input files are in DOS file format. Contact the *inSALMO* developers if the problem persists.
When I run a large *inSALMO* experiment, the model runs successfully for a long time and then suddenly crashes with an error saying something about "xmalloc". Re-running it produces the same error in the same place.	Unfortunately, this is due to the limited amount of random-access memory that Windows can allocate to any one program and the way that RAM gets fragmented as the model runs. The only solutions are to run smaller jobs, or run the model in a 64-bit Linux version of the model.

Appendix C: Graphical User Interface Guide

inSALMO Version 1.0
Graphic User Interface Guide

Prepared by:

Steven F. Railsback
Lang, Railsback & Associates
Arcata, CA
www.LangRailsback.com

Colin Sheppard
Arcata, CA

Last changed: October 10, 2011

Contents

1. Introduction ... 6
1.1. Document objectives ... 6
1.2. Document format ... 6

2. Terminology ... 6

3. Basic Tasks ... 7
3.1. Which files are where? ... 7
3.2. Installing the GUI and *inSALMO* ... 8
3.3. Starting the GUI ... 8
3.4. Using the help files ... 8
3.5. Opening a project ... 8
3.6. Saving and closing a project ... 9
3.7. Saving a project as a new project ... 9
3.8. Deleting a project ... 9
3.9. Finding errors and warnings ... 9

4. Creating Projects ... 9
4.1. When to create a new project ... 10
4.2. Creating a new project ... 10

5. Adapting the New Project to an Application ... 11
5.1. Habitat reaches ... 11
5.1.1. Adding or removing reaches ... 11
5.1.2. Editing a reach's setup ... 12
5.1.3. Reach habitat parameters ... 12
5.2. Species or races ... 13
5.2.1. Adding or removing a species ... 13
5.2.2. Editing species setup information ... 13
5.2.3. Salmon parameters ... 14
5.3. Model run settings and output file names ... 14

6. Setting Up Experiments ... 15
6.1. Why to always check the Experiment Interface ... 15
6.2. General information on experiments ... 15
6.3. Setting up an experiment ... 16
6.3.1. Set the number of scenarios ... 16
6.3.2. Set the number of replicates ... 17
6.3.3. Select parameters and their values ... 17
6.4. Removing parameters from an experiment ... 18

7. Executing and Observing Model Runs ... 18
7.1. Batch vs. graphics modes ... 18
7.2. Terminating a run ... 19
7.3. Changing Observer setup values ... 19

8. Analyzing Output ... 19
8.1. Purpose of the View Results tool ... 19
8.2. Using the tool ... 20
8.3. Additional outputs ... 21

9. The Limiting Factors Tool 21

10. Changing Defaults.....22

10.1. Default parameter values and recommended ranges 22

10.2. The default project.....22

10.3. Limiting Factor Tool default parameters 22

11. Troubleshooting 23

1. Introduction

1.1. Document objectives

This digital document provides user instructions and guidance for the graphical user interface (GUI) for version 1.0 of *inSALMO*. The GUI is an optional software tool for setting up, executing, and observing *inSALMO* simulations. The GUI was developed along with version 1.0 of *inSALMO* as part of the project "Improvement of Salmon Life-Cycle Framework Model (inSALMO)" conducted by Lang, Railsback and Associates (LRA) and USDA Forest Service, Pacific Southwest Research Station, for the US Bureau of Reclamation.

This document is packaged with the GUI as a help file. Users will also need to be familiar with the contents of three other documents that are also provided as help files:

- The Model Description provides all the detailed methods, assumptions, parameter values, etc. defining what *inSALMO* does.

- The Software Documentation describes in detail the input files needed to run the model, and the graphical and file outputs it produces.

- The Limiting Factors Tool Documentation describes the part of the GUI that automates experiments to evaluation the strength of different factors that could limit salmon populations.

1.2. Document format

This help file is organized around the different tasks that users typically conduct in using *inSALMO*. It therefore consists of many short sections that each tell how to do one task. However, these are organized into categories corresponding to the major phases of a model application: setting the model up for a new site or application, setting up simulation experiments, executing the model, and analyzing results. Hence, users typically will use the help file by looking up what they are trying to do and reading how to do it.

The index is designed to support users trying to understand some particular function or part of the GUI. When in need of help for how some particular button or tab works, the index is where to look.

2. Terminology

This document and the GUI use the following specific definitions. Users should also be familiar with the terms and conventions used in the model itself, defined in Section 2 of the model description document.

Application: A modeling analysis of a particular question about a particular site. Examples include (a) calibrating outmigration timing and size at the Clear Creek 3A and 3C reaches, and (b) predicting changes in the number of large outmigrants at Clear Creek due to three alternative instream flow scenarios. Typically there is one **project** for each application.

Project: All the computer files needed to apply *inSALMO* to a particular study site and question. Behind the GUI, a project is simply a directory (folder) of the input, control, and output files for

an application. A project's name is the same as the name of the directory it resides in. Separate projects can be created for separate study sites, or for separate analyses at the same site.

Parameter: In the GUI, "parameter" refers to any variable that appears in the model's setup and parameter files. Hence, parameters can include variables that control the model's input and execution (e.g., input file names; names of habitat reaches and fish species) as well as the habitat and fish parameters that are parameters in the conventional sense of coefficients for the model's equations.

3. Basic Tasks

3.1. Which files are where?

When *inSALMO* is installed on a Windows computer, all its components reside in a directory tree that can be located anywhere. Typically the model is installed to the "My Documents" folder so that its directory tree starts at `C:\Documents and Settings\(USER NAME)\My Documents\inSALMO_1-0`. The remainder of this section refers to directories under this `inSALMO_1-0` directory, no matter where the model is installed.

The files that users need to be aware of are:

- The actual GUI program is an executable file at `\inSALMO_1-0\GUI\inSALMO-GUI.exe`. (The actual name may be slightly different, as different versions may be identified by including their release date in the file name.)

- The *inSALMO* model executable is at `\inSALMO_1-0\Code\inSALMO\.libs\insalmo.exe`.

- Projects can actually reside anywhere on the computer. By default each project is in a separate subdirectory of `\inSALMO_1-0\Projects`. A built-in project packaged with *inSALMO* is at `\inSALMO_1-0\Projects\ClearCreekDefault`.

- Help files are at `\inSALMO_1-0\Help`. (The help files are accessed via the GUI menu but can also be used by clicking on them from Windows Explorer. They can also be updated by simply replacing them with new versions.)

This diagram illustrates the basic structure of the *inSALMO* file system but omits many subdirectories that users need not be familiar with:

```
\inSALMO_1-0
            \Code\inSALMO
            |
            \Help
            |
            \GUI
            |
            \Projects\ClearCreekDefault
                     |
                     \ClearCreekProj2
                     |
                     \(etc.)
```

3.2. Installing the GUI and *inSALMO*

The graphical interface, the *inSALMO* model, and example input files are distributed in a zip archive file, typically named something like `inSALMO_1-0.zip`. The model is installed in two simple steps:

- Extract the zip file to any directory (including your "My Documents" folder). This will create the `\inSALMO_1-0` directory tree underneath the directory where it is extracted. If using Windows' build-in unzip utility, make sure it extracts *all* the files.

- In the Windows Explorer, right-click on the GUI executable file `\inSALMO_1-0\GUI \inSALMO-GUI.exe`. Select "Create Shortcut" to create a shortcut file. Drag this new shortcut file to your desktop.

3.3. Starting the GUI

To launch the *inSALMO* GUI: Simply double-click on the inSALMO-GUI shortcut on the desktop. (Alternatively, you can double-click on `\inSALMO_1-0\GUI\inSALMO-GUI.exe` in Windows Explorer.)

3.4. Using the help files

To use the help files packaged with the GUI: Open any of the four help files by clicking on the "Help" item on the GUI's main menu. This opens a list from which the help files can be selected. Separate files are provided for the GUI (this document), the model's software, and a description of the model's formulation. Each help file includes an expandable table of contents, an index, and a search function.

3.5. Opening a project

To open an inSALMO project: Either click on the button labeled "Open", or use the GUI menu selections Project > Open. These let you browse to select a project by selecting the directory (folder) containing the project. You can only select a directory, not a file.

3.6. Saving and closing a project

To save changes to a project: Changes to a project must be saved before executing it. Either click on the button labeled “Save”, or use the GUI menu selections Project > Save. You can save the project’s changes when closing it.

To close an inSALMO project: Click on the GUI menu selections Project > Close. If you have unsaved changes, you will be prompted to save them or quit without saving the changes.

3.7. Saving a project as a new project

To save a project with a new name, as a new project: Click on the GUI menu selections Project > Save as. This takes you to a dialog where you can simply type in a new project name, and the GUI will save the project in a new directory with that name, under the `\inSALMO_1-0\Projects` directory. You can also browse to a different directory if you want to save the new project elsewhere. Other options are to:

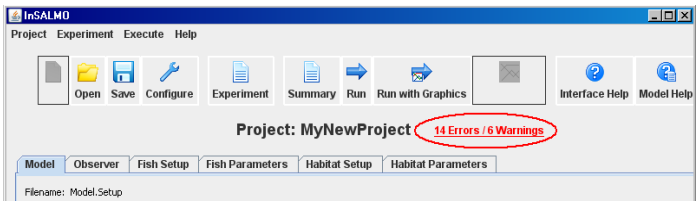
- Select an existing directory, in which case any project in it will be overwritten, or
- Select the project’s current directory, which is the same as simply saving the project and does not create a new project.

3.8. Deleting a project

There is no function in the GUI to delete projects. Instead, you can simply delete the project directory, for example by using Windows Explorer.

3.9. Finding errors and warnings

The GUI identifies obvious input errors and warnings about questionable parameter values. The number of errors and warnings present at any time appears in red text near the project title (illustrated below).



Clicking on this red text opens a table showing all the errors and warnings in the project. Clicking anywhere in this table takes you to where the the error or warning occurs, where it is flagged for you to correct.

4. Creating Projects

This section discusses when to create a new project instead of just modifying an existing one, and describes the three alternative ways to create a new project.

Because *inSALMO* uses many input files, some of which depend on each other, the process of creating a new project is complex. New projects are therefore easiest to create by modifying existing ones; *inSALMO* is packaged with a “template” project to facilitate this.

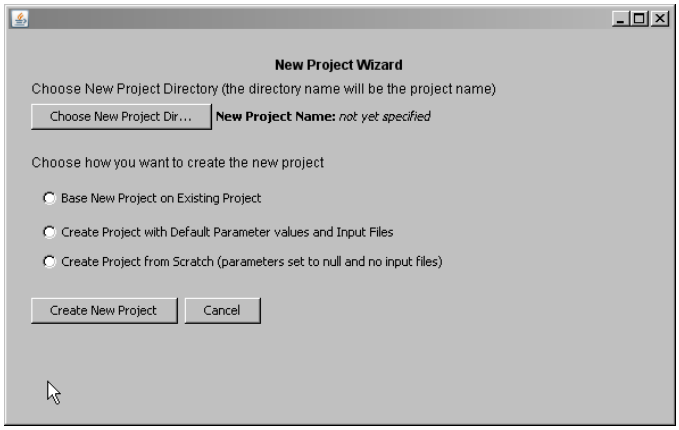
4.1. When to create a new project

A new project should be created whenever you want to create a new application or analysis and keep it separate from previous applications. Keep in mind that any time you change a project or re-run *inSALMO*, previous results for the project will be destroyed. Examples of when to create a new project include:

- When modeling a new study site, or making substantial changes to an existing site (e.g., adding another reach);
- When starting a new analysis, or kind of analysis (calibration, limiting factors study, comparison of management alternatives) that you want to save separately from previous analyses;
- When model input changes substantially (e.g., a new year is added to the input set; spawner initialization data are revised); and
- When you want to play around with the model, making experimental or temporary model runs, without destroying previous results or worrying about documenting or un-doing the changes you make. Just save the project as a new project that you can delete when done.

4.2. Creating a new project

The GUI includes a new project wizard that provides three alternative ways to create a new project. Start the wizard by selecting Project > New from the main menu.



Click the “Choose New Project Directory” to select a directory location (which is normally just a new subdirectory of `\inSALMO_1-0\Projects`) and name. The new directory name becomes the project name.

You then choose from three alternative ways to create project:

- Base new project on an existing project, which copies an existing project and opens it as the new one. This has the same effect as saving the existing project as a new one (Section 3.7).

- Create project with default parameters and inputs, which copies *inSALMO*'s default project as the new project.

- Create project from scratch, which requires you to input all parameter values, input file names, etc.

Finally, click the Create New Project button to finish creating the project. The steps at Section 5 guide the process of completing a new project once it has been created.

5. Adapting the New Project to an Application

Once you have created a new project, the next step is to adapt it to the application you want to conduct. If you are creating an entirely new study at a new site, the process can be extensive and typically requires many iterations of finding and correcting errors. But this process can be very simple if you are just conducting a new analysis of an existing application.

The process of adapting a project to a new application is a matter of updating or replacing the model's configuration (number of reaches or their configuration; number of species), inputs (flow, temperature, or turbidity files; cell data files), or parameter values. Here is a guide to what you might need to change.

- If you are just running new simulation experiments that do not require new input files, you can just use the Experiment Manager (Section 6) instead of making any of the permanent changes described in this section.

- If your application requires new flow, temperature, or turbidity input files, see Section 5.1.2.

- If you are applying the model to a new site for the first time, then you will need to carefully update all the inputs. To do so, go through the steps in sections 5.1 through 5.3 carefully, consulting the software guide to understand each setup, parameter, and input file. Use the GUI's error and warning checking (Section 3.9) to help find and correct mistakes.

These kinds of changes are made through the GUI's Configure Interface, which is accessed by clicking on the Configure button.

5.1. Habitat reaches

The reach setup information defines how many reaches are in a project, how those reaches are linked (which is downstream of which, etc.), and which files contain the input needed for each reach. This information corresponds to the Reach Setup file described in the inSALMO software guide.

5.1.1. Adding or removing reaches

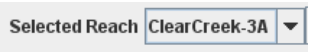
To add a reach from the project: Click on the Habitat Setup tab. (If this tab is not visible, click the Configure button.) Near the top of this tab is a box labeled Add Reach. Click on that box, and choose whether to add a reach by copying an existing one or creating one with default values.

To remove a reach from the project: Click on the Remove Reach box in the Habitat Setup tab, and select which reach to remove.

5.1.2. Editing a reach's setup

Each reach is defined by setup information providing its name, input file names, and junction numbers.

To edit the reach's setup information: Click on the Habitat Setup tab. (If this tab is not visible, click the Configure button.) Use the selector near the top of the tab to select the reach you want to edit. The reach selector looks like this:



Each of the reach's setup parameters is displayed in a table. The first column describes the parameter, and the third column contains the name of the corresponding variable in the inSALMO software. The second column provides the parameter's current value. Double-click on the values in the second column to change them.

If the box turns red, you entered an invalid value (e.g., an alphanumeric character string for a floating point variable). If the box turns yellow, you entered a value outside the parameter's normal range (as defined by the model's authors); the value will be accepted and used, but you should check it carefully.

The input files providing cell variables, hydraulic data, and daily flow, temperature, and turbidity input are identified as part of habitat setup for each reach. These daily values cannot be edited from within the GUI but only by editing the individual input file. Simulation experiments to compare alternative flow and temperature regimes require multiple input files for these variables. How to set up such experiments is described in the Experiment Interface discussion at Section 6.3.3.

To select input files: Some of the parameters in the reach setup are the names of input files for the reach. Clicking on their value in the parameter table opens a dialog that lets you browse to select a file.

- Selecting a file copies that file into the project directory; you can find a file anywhere on your computer or network and it will be permanently copied into your project directory.

- You are responsible for making sure the file is in the proper format. *inSALMO* does some error checking when it executes, but the GUI does no checking of whether selected files will work.

5.1.3. Reach habitat parameters

This step corresponds to editing a reach's habitat parameter file.

To edit the parameter values for a reach: Click on the Habitat Parameters tab. (If this tab is not visible, click the Configure button.) Use the selector near the top of the tab to select the reach you want to edit.

The reach's habitat parameters are displayed in a table. The first column describes the parameter, the third column contains the name of the corresponding variable in the *inSALMO* software, while the second column provides the parameter's current value. Double-click on the values in the second column to change them.

If the box turns red, you entered an invalid value (e.g., an alphanumeric character string for a floating point variable). If the box turns yellow, you entered a value outside the parameter's normal range (as defined by the model's authors); the value will be accepted and used, but you should check it carefully.

5.2. Species or races

Salmon species (or race, such as fall-, late fall-, and spring-run Chinook, which are treated as species) can be added to or removed from a project, but doing so requires corresponding changes in the *inSALMO* software; see the software guide for how to do so. Once an additional species is in the software, it must then be included in a project via the GUI as described here. *Simply adding a species to the project as described here, without also modifying the inSALMO software, does **not** add a new species to the model!* Instead, it will just cause the model to fail when executed.

5.2.1. Adding or removing a species

To add a species: Click on the Fish Setup tab. (If this tab is not visible, click the Configure button.) Click the Add Species box near the top of the tab. The dialog that pops up lets you select between copying an existing species (and later editing its setup and parameter values) our using the default species values. The dialog is also where you enter the name of the new species. Remember that the new species must be added to the *inSALMO* code, and that the species name given here must be exactly the same as in the code.

When the software includes more than one species, the number of species in a project can be reduced by removing species here in the GUI. (It is OK for a species to exist in the *inSALMO* code but not in the GUI project.)

To remove a species from the project: Click on the Fish Setup tab. (If this tab is not visible, click the Configure button.) Near the top of this tab is a box labeled Remove Species. Click on that box, which lets you select which species to remove.

(These steps correspond to editing the Species setup file.)

5.2.2. Editing species setup information

Each species has several setup variables: the name of the input file for spawner initialization and the display color.

To edit a species' setup information: Click on the Fish Setup tab. (If this tab is not visible, click the Configure button.) Use the selector near the top of the tab to select the species you want to edit.

The species setup parameters are displayed in a table. The first column describes the parameter, and the third column contains the name of the corresponding variable in the

inSALMO software. The second column provides the parameter's current value. Double-click on the values in the second column to change them.

If the box turns red, you entered an invalid value (e.g., an alphanumeric character string for a floating point variable). If the box turns yellow, you entered a value outside the parameter's normal range (as defined by the model's authors); the value will be accepted and used, but you should check it carefully.

Spawner initialization data (the number, size, etc. of spawners for each reach, year, and species) are provided via an input file identified as part of the fish setup information. This input cannot be edited from within the GUI, but only by editing the spawner initialization data file itself.

To select input files: Some of the parameters in the species setup are the names of input files. Clicking on their value in the parameter table opens a dialog that lets you browse to select a file.

- Selecting a file copies that file into the project directory; you can find a file anywhere on your computer or network and it will be permanently copied into your project directory.

- You are responsible for making sure the file is in the proper format. The GUI does no checking of whether selected files will work. (However, the *inSALMO* code does some error checking of input when you execute it; see the Software Guide.)

5.2.3. Salmon parameters

The parameter values used by each species of salmon can be edited via the GUI. These edits permanently change the parameter values; to make temporary changes in simulation experiments, see Section 6. (This step corresponds to editing a salmon parameter file.)

To review and edit salmon parameters: Click on the Fish Parameters tab. (If this tab is not visible, click the Configure button.) Use the selector near the top of the tab to select the species you want.

The fish parameters are displayed in a table. The first column describes the parameter, and the third column contains the name of the corresponding variable in the *inSALMO* software. The second column provides the parameter's current value. Double-click on the values in the second column to change them.

If the box turns red, you entered an invalid value (e.g., a floating point number for a date variable). If the box turns yellow, you entered a value outside the parameter's normal range (as defined by the model's authors); the value will be accepted and used, but you should check it carefully.

5.3. Model run settings and output file names

Model parameters control the dates when simulations start and end, the random number generator seed, and output file names. (These parameters are in the Model Setup file.)

To change model parameter values: Click on the Model tab. (Click on the Configure button if this tab is not visible.) You will then see a table of model run parameters; edit the values in the second column. Invalid values will result in their box turning red.

6. Setting Up Experiments

Setting up and running simulation experiments should be the most common use of the *inSALMO* GUI. Once the model has been applied to a site, simulation experiments are how the model is used to address management and research problems.

Simulation experiments include designing a set of model runs to address a particular question (e.g., how does outmigration success vary with fish predation risk? with instream flow?), executing the runs, and analyzing the results. This section addresses setting up and executing the model runs.

The GUI's Experiment Interface is accessed by clicking on the Experiment button. (The tasks described in this section all correspond to controlling *inSALMO*'s Experiment Manager via the Experiment Setup file, which is described in the Software Guide.)

6.1. Why to always check the Experiment Interface

The Experiment Interface can control selected model inputs and parameters, and it overrides the project's standard values for those inputs and parameters. Hence, to know exactly what input the model is using, you must look at the Experiment Interface. Otherwise, you might not realize that the Experiment Interface is causing the model to use input other than what you expect.

If you just want to run the model with only standard input files and parameter values, click on the Experiment button. On the Control tab is a "Clear Experiment" button clears out the Experiment Interface so it has not effect. After clicking this button the number of scenarios is 1, the number of replicates is 1, and, on the Parameters tab, there are no parameters added to the experiment.

6.2. General information on experiments

The purpose of the Experiment Interface is to allow users to set up and execute simulation experiments that use multiple model runs with different inputs, without having to modify other parts of their project. With a few simple menu selections, a complex experiment can be set up to run automatically. The experiments can include both <u>scenarios</u> and <u>replicates</u>.

- A scenario is a single set of inputs and parameter values; different scenarios are defined by specifying which inputs or parameters differ among them.

- Replicates are repeated runs of the same scenario, with only the pseudorandom numbers (which primarily affect mortality) differing among replicates. Replicates are typically used to see how much of the variation in model results among scenarios is due only to the model's stochasticity.

The Experiment Interface can be used to set up a simulation experiment with multiple scenarios and replicates; it then can execute the experiment by running *inSALMO* repeatedly while automatically changing the model's parameters or inputs for each scenario and replicate. For example, the user may set up an experiment with 10 scenarios (perhaps five different values for some parameter, for each of two flow input files) and five replicates. The Experiment Manager would determine that 50 model runs are needed and start one run after another. At the start of each model run the Experiment Interface stops the model after parameter values have been read in, then overwrites the value of any parameters that are part of the experiment. For

replicates, the Experiment Interface simply changes the random number seed. The Experiment Interface then re-starts the model and has no more effect until the next run starts.

The Experiment Interface can be used to set up “factorial” experiments in which two or more parameters are varied, and all combinations of each parameter are executed. However, you must set the experiment up manually; the Experiment Interface cannot automatically generate all combinations of parameter values. For example, a calibration experiment might include four values of drift food concentration (habDriftConc = 1.0, 2.0, 3.0, and 4.0E-10) and three values of aquatic predation risk (mortFishAqPredMin = 0.95, 0.97, 0.99). The factorial experiment requires the user to set up 12 scenarios:

Scenario	habDriftConc	mortFishAqPredMin
1	1.0E-10	0.95
2	1.0E-10	0.97
3	1.0E-10	0.99
4	2.0E-10	0.95
5	2.0E-10	0.97
6	2.0E-10	0.99
7	3.0E-10	0.95
8	3.0E-10	0.97
9	3.0E-10	0.99
10	4.0E-10	0.95
11	4.0E-10	0.97
12	4.0E-10	0.99

Below is general guidance for setting up an experiment. The Experiment Manager section of the *inSALMO* Software Guide provides detailed examples of many kinds of experiment; users are referred to that section for an understanding of how the Experiment Interface is typically used and how to set up common experiments.

6.3. Setting up an experiment

Setting up an experiment includes the following three steps.

6.3.1. Set the number of scenarios

The first step is determining how many scenarios the experiment has. Each scenario is one combination of the parameter values, input files, etc. that are varied in the experiment.

To set the number of scenarios in an experiment:

- Click on the Experiment button if necessary to enter the Experiment Interface.
- Click on the Control tab, then the box in the second (“Value”) column of the parameter table, in the row labeled “Number of Scenarios”.
- Enter the number of scenarios in the box.

The number of scenarios can be as low as 1 (which is common if the purpose of the experiment is only to execute replicates with a project’s standard input values). Otherwise, users must calculate the number of parameter and input combinations in their experiment.

6.3.2. Set the number of replicates

Adding replicates to an experiment simply causes each scenario to be repeated with a new random number generator seed. (Replicates can be used even if no scenarios other than the standard input are defined. Just change the number of replicates and add nothing else to the experiment. Likewise, it is common to use just one replicate when running many scenarios.)

To set the number of replicates in an experiment: Click on the Control tab of the Experiment Interface. Click on and edit the box containing the value of the parameter “Number of Random Replicates”.

6.3.3. Select parameters and their values

The third step is to identify the parameters to vary among the scenarios, and set their value for each scenario. This step is not necessary if the experiment only includes replicates of the standard input values.

To select experiment parameters and their values:

- Click on the Parameters tab of the Experiment Interface. Initially this tab will be blank except for the “Add Parameter” and “Remove Parameter” buttons.

- Click on the Add Parameter button. In the dialog that opens, select the kind of parameter to include in the experiment; the options are input filenames, output filenames, model parameters, fish parameters, and habitat parameters.

- After selecting the kind of parameter, click on the box labeled “Select the parameter” to select from a list of the available parameters; the list depends on which kind of parameter is being chosen.

- If the parameter is an input filename, a fish parameter, or a habitat parameter, there is a second box labeled “Select the instance name”. This box lets you choose whether the experimental variation in the parameter’s value applies to all instances (habitat reaches; fish species) or just to one reach or species. Select “NONE” for the instance name if you want the experiment to affect all instances.

- Click the dialog’s “Add Parameter” button to close it and return to the Parameters tab.

- The Experiment Parameters table on the Parameters tab now includes the selected parameter. The table also includes information on which code class the parameter belongs to, which reach or species the experiment applies to, and what type of value the variable is; you do not need to understand and should not edit this information.

- The Experiment Parameters table also includes one row labeled “Value” for each scenario. (If the experiment was given 15 scenarios in the first step, Section 6.3.1, then there will be 15 “Value” lines.) You must now manually enter the parameter’s values for all scenarios. Click on the value boxes in the second column and add the new values. The box will turn red if an invalid value type is entered, and yellow if the value is outside the range of recommended values.

- In revising an experiment, the number of scenarios can be changed on the Control tab. In that case, use the Experiment Parameters table's buttons labeled "Remove Value" and "Add Value" to change the number of values for each parameter.

6.4. Removing parameters from an experiment

In revising an experiment, you may choose to remove a parameter from those controlled by the Experiment Interface.

To remove a parameter from an experiment: Click on the Parameters tab of the Experiment Interface. Click on the "Remove Parameter" button and select the parameter to remove from the dialog that pops up.

7. Executing and Observing Model Runs

This section explains how to execute *inSALMO* model runs once they are set up. When the GUI executes *inSALMO*, it always executes whatever experiment has been defined in the Experiment Interface.

7.1. Batch vs. graphics modes

inSALMO can be executed in one of two modes.

Graphics mode execution runs the model with its graphical displays, which are updated after each simulated day. These graphics are described in the Software Guide. When a model run is started, the main control panel opens, and the user must click on "Start" ***two times***. If an experiment with multiple model runs is set up, the model stops at the end of each run, so the user must click "Start" on the main control panel again to start the next run. (This control panel is not visible from the GUI; it appears when the first run starts.) Graphics mode is usually used when setting up new projects to check and observe the model's behavior.

The main control panel

Batch mode execution runs the model with no output other than a line written to the GUI's output window each simulated day to report what scenario, replicate, and date was just simulated. (This output window appears on the GUI's Summary Interface when a model run is started.) In batch mode, all the model runs in an experiment are executed sequentially and automatically, without stopping. Batch mode is therefore typically used for simulation experiments. (*inSALMO* executes substantially faster in batch mode.)

To start the model in graphics mode: When ready to execute the model, click the GUI's "Run with Graphics" button. The GUI displays its Summary Interface, and a the main control panel opens. Click the main control panel's "Start" button two times. The model's other graphical controls and displays (as described in the Software Guide) then open and the first model run starts.

To start the model in batch mode: When ready to execute the model, click the GUI's "Run" button. The GUI switches to its Summary Interface, and execution starts immediately.

7.2. Terminating a run

Model runs can be stopped before they are complete by clicking on the "Terminate Model Run" button at the bottom of the Summary Interface.

7.3. Changing Observer setup values

The user can control several characteristics of *inSALMO*'s graphical displays, such as whether cell depth or velocity is displayed, and how big the animation raster is. These controls are described in the Observer Setup section of the Software Guide.

To change Observer setup parameters: Click on the Configure button to switch to the GUI's Configure Interface. Click on the Observer tab, and edit values in the second column of the parameter table.

8. Analyzing Output

The GUI provides a tool for opening key model results in Excel and rapidly viewing summary tables and graphs from simulation experiments. This tool is accessed by clicking on the "View Results" button. This section describes this tool and other methods for analyzing output.

8.1. Purpose of the View Results tool

Individual-based models such as *inSALMO* produce many kinds of output and hence can be "viewed" and analyzed in many ways; imagine a real fish population from which data could be obtained with almost no cost and no uncertainty. Clearly, deciding how much output to analyze in how many ways requires a tradeoff: looking at too much output can be overwhelming and confusing, while looking at too little does not take advantage of the model's most important capability: providing explanation of what processes and factors are most important, given the model's assumptions and a particular set of inputs. Users are certainly discouraged from blindly believing model results; instead, results should be scrutinized to determine *why* they arose and how believable they are.

The View Results tool is intended to provide a rapid overview of *inSALMO*'s most important results and a quick comparison among the scenarios included in a project's simulation experiment. The results it looks at are:

- The total number of outmigrants in each scenario (and the variation among replicates if they are used);
- The number of large (length > 5 cm) outmigrants;

- The number of eggs produced by spawners and the number of fry hatching from redds; and

- The causes of egg mortality.

The tool is designed to take advantage of Excel's powerful PivotTable and PivotChart facilities, which can break out and summarize data easily and flexibly. The View Results tool imports key results to Excel and updates a number of built-in PivotTables and PivotCharts. The Tool will be much more useful if you are familiar with these Excel facilities.

8.2. Using the tool

The View Results tool can be used any time *after* a model run or simulation experiment has been executed fully. The tool displays *results of the last model run or experiment executed*. If a simulation experiment was conducted then its project saved, the project can be re-opened in the GUI and the View Results tool used to look at the experiment's results without re-running the experiment. (This is one reason you are encouraged to save each major simulation experiment in a separate project.)

Once a project's simulations have been executed, you can click on the "View Results" button. The button causes several things to happen. First, the GUI opens an analysis workbook in Excel; this workbook is called Analysis_Setup.xlsxm (the extension ".xlsm" means it is a macro-enabled workbook). This workbook contains macro code that automatically imports two output files produced by *inSALMO* the last time the project was executed. These are the outmigrants output file (typically named something like Outmigrants_Out.csv) and the redd output file (typicalled named something like Redds_Out.csv). **This Excel import process will very likely halt, with Excel displaying a "Security warning" saying that macros have been disabled (**Figure 1**). You must choose the option to allow macro content for the analysis to proceed.**

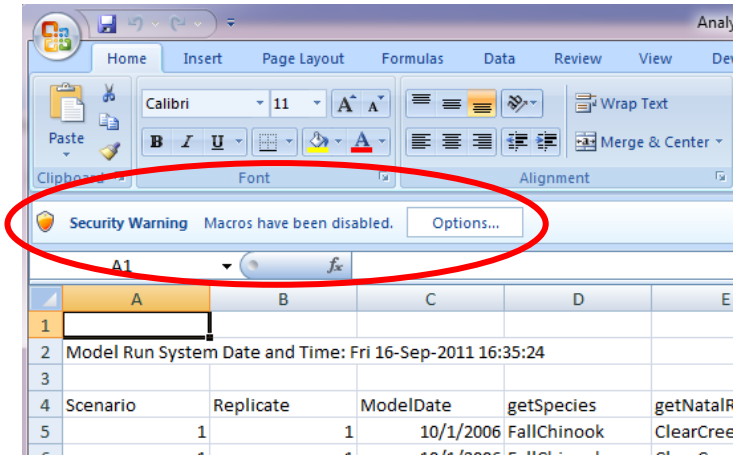


Figure 1. Macros must be enabled manually by clicking on the "Options" button for the View Results Excel workbook to import model results and display results.

Next, the workbook code automatically updates its results tables and graphs. Finally, the user is prompted to save the workbook as a new, standard Excel file (with macros disabled; extension .xlsx).

The analysis workbook can then be examined and customized as desired. The workbook contains separate analyses on separate pages; these include summary analyses such as:

- Tables with the number of total and large outmigrants broken out by scenario and replicate (so that the mean and standard deviation across replicates can be calculated for each scenario);
- Bar charts showing the number of total and large outmigrants by scenario;
- Tables with the number of redds and emergent fry broken out by scenario and replicate;
- Bar charts showing how many eggs died of each cause, by scenario; and
- A graph showing when redds were created.

Additional analyses are easily created from the raw results in the spreadsheet.

8.3. Additional outputs

Users of the GUI are not restricted to examining only the outputs in the View Results tool. The model also produces detailed results on the number and size of live fish (in each reach, and broken into several size classes), and the number of fish that died of each kind of mortality. These results are not included in the View Results tool because they are extensive and complex, and not always important for understanding simulation experiments.

These additional outputs can also be somewhat confusing; for example, the number of dead fish can often appear high not because conditions suddenly became more risky but just because more live fish were created. Because some percent of juvenile salmon always die, and this percent can be density dependent, more spawning success inevitable produces more fish that die before outmigration.

Live fish censuses and dead fish counts are available from the “Population output file” and “Mortality output file”, the names of which are specified in the Model parameters table (Section 5.3). These files are written in .csv format and open readily in Excel. They can be found in the project directory.

There are a number of optional output files that *inSALMO* can be modified to produce. These files, and the other output files described here, are documented in the Software Guide help file.

9. The Limiting Factors Tool

The *inSALMO* GUI includes a button to access the Limiting Factors Tool, which automates simulation experiments to analyze the relative strength of various factors that can affect salmon populations. This tool is documented separately in its own help file.

10. Changing Defaults

The GUI can be customized by the various kinds of default values it provides. Changing these defaults is simply a matter of finding and editing the correct file.

10.1. Default parameter values and recommended ranges

The GUI provides default values for all *inSALMO* parameters, and recommended ranges of acceptable values for some parameters. These default values and ranges are stored in plain text files in the subdirectory `inSALMO_1-0\GUI\metaprojectdata`. These metaprojectdata files have names describing their function: the file name includes “Param” or “Setup” for whether they provide default parameter values or setup information, and “exp”, “hab”, etc. for whether they apply to the Experiment Manager (Section 6), habitat (Section 5.1), the model (Section 5.3), the observer (Section 7.3), or fish species (Section 5.2). (The file `lftSetup.txt` is discussed below at Section 10.3.)

The default parameter values and ranges are changed by simply editing the appropriate metaprojectdata file. The files are in plain text with values separated by tab characters. The files can be edited either with a text editor (Notepad, etc.) or imported to Excel and exported as tab-separated plain text. The metaprojectdata files are self-explanatory with the exception of a variable “DisplayOrder” for each parameter. This variable simply controls the order in which parameters appear when displayed in the GUI.

If a new parameter is added to the model, it is added to the GUI simply by adding it to the appropriate metaprojectdata file. For example, if *inSALMO* is modified so that fish require a new parameter fishRespirationParamZ, then this parameter is added to the GUI by editing the file speParam.txt to include add the parameter, its default value, its description, display order, variable type, and (optionally) lower and upper bounds on its recommended values.

The default parameter values are used only when creating a new project from scratch (Section 4.2). When a project is created by copying an existing project, including the default project, all its parameters are copied from the source project.

10.2. The default project

New projects can be created by opening and editing a default project (Section 4.2). This default project is stored in the directory `inSALMO_1-0\GUI\DefaultProject`. It can be revised or replaced by editing or replacing the model input files in that directory.

10.3. Limiting Factor Tool default parameters

The Limiting Factors Tool (LFT; Section 9) uses a number of parameters, and their default values are stored in the file `inSALMO_1-0\GUI\metaprojectdata\lftSetup.txt`. These defaults can be updated and changed the same way that other default parameters are (Section 10.1).

Note that the default parameters for the LFT are used only when a new project is created from scratch (Section 4.2). When a project is created by copying an existing project, or from the default project, its LFT parameters are copied from the source project. These parameters are in a file `LimitingFactorsTool.Setup` in the project directory.

11. Troubleshooting

If you have problems installing or running the GUI, some things to check are:

- Make sure that all the files and directories in the zip file used to distribute *inSALMO* and the GUI were extracted. In some cases, the Windows unzip utility does not by default extract all files.

- Check for confusion among the different “My Documents” directories that can be on one computer. On some computers there can be separate “My Documents” directories for “All Users” as well as for the specific individual user(s), and it is easy to extract the model into the wrong one.

If you have problems getting *inSALMO* to run when you click the “Run” or “Run with Graphics” buttons, see the extensive Troubleshooting Guide at the end of the Software Documentation help file.

Note that error messages from the *inSALMO* code appear in the “Model Execution Output” window on the Summary Interface. These error messages can be extremely important and helpful.

Appendix D: Limiting Factors Tool Guide

inSALMO Version 1.0
Limiting Factors Tool Guide

Prepared by:

Steven F. Railsback
Lang, Railsback & Associates
Arcata, CA
www.LangRailsback.com

Colin Sheppard
Arcata, CA

Last changed: October 10, 2011

Table of Contents

1. Objectives and Overview of the Limiting Factors Tool 6
1.1. What are “limiting factors”? 6
1.2. The approach: sensitivity analysis with parameter uncertainty 6
1.3. Interactions among factors 7
1.4. When to use the LFT 7
1.5. LFT parameters 7

2. Limiting Factor Experiments 7
2.1. Base flow 8
2.2. Food availability 8
2.3. Winter water temperature 8
2.4. Summer water temperature 9
2.5. Spawning gravel availability 9
2.6. Velocity shelter availability 10
2.7. Hiding cover availability 10
2.8. Piscivory risk 11
2.9. Redd scouring 11
2.10. Number of spawners 12

3. Uncertainty Consideration 12

4. Results Analysis and Interpretation 14
4.1. Analysis methods and interpretation of results 14
4.2. Results formats 14

5. Parameters Controlling the Limiting Factors Tool 15

6. Using the Tool 19

7. Computational Considerations 22

LFT Parameter Index 23

1. Objectives and Overview of the Limiting Factors Tool

The Limiting Factors Tool (LFT) is a software add-on to the *inSALMO* individual-based salmon model. Its general purpose is to automate simulation experiments that address an important habitat management question: which manageable habitat factors have high potential for enhancing fish populations?

1.1. What are "limiting factors"?

"Limiting factors" is a common but not completely accurate term for environmental variables or processes that strongly affect a fish population. A traditional notion of limiting factors is that at any one time there is one factor that "limits" a population from growing; typically, food availability, predation pressure, reproductive output, and habitat space are considered potential limiting factors. Improving the limiting factor is expected to increase the population while the "non-limiting" factors are expected to have little effect.

In contrast, individual-based models such as *inSALMO*, in which the individual fish have adaptive behaviors to trade off growth and predation risk, predict that multiple factors can have strong effects on population growth at any one time, and there is not necessarily a threshold above (or below) which a factor such as food availability has no effect[1]. A more useful way to think about limiting factors, at least in the context of *inSALMO*, is: how strongly do various environmental factors affect the population, and which factors offer high potential for population enhancement?

The LFT was designed to evaluate a number of habitat factors considered especially important to California Chinook salmon populations. In some cases, other factors may seem important to evaluate; when this occurs, experiments similar to those performed automatically by the LFT can usually be conducted using *inSALMO*'s Experiment Manager (see the model's software documentation) and parameters or inputs that control those factors. However, *inSALMO* and the LFT can only evaluate factors and their effects if those effects are explicitly represented in *inSALMO*. Some potentially important limiting factors such as siltation of spawning gravel or angler harvest of adults are not represented in version 1.0 of *inSALMO* and hence cannot be evaluated with the LFT.

1.2. The approach: sensitivity analysis with parameter uncertainty

The LFT addresses its objective by implementing a sensitivity analysis approach. It includes built-in experiments for habitat variables that (a) likely have—or are often believed to have—strong effects on the salmon life stages represented in version 1.0 of *inSALMO*, and (b) potentially could be changed via management actions. Each LFT experiment runs the model repeatedly using a wide range of values for one factor. The tool then computes and compares the degree to which the model's key output—the number of relatively large outmigrating juveniles—responds to each factor.

---

[1] Railsback, S. F., and B. C. Harvey. In press. Importance of fish behaviour in modelling conservation problems: food limitation as an example. Journal of Fish Biology. doi:10.1111/j.1095-8649.2011.03050.x

Results of the limiting factors experiments depend on the values used for *inSALMO*'s parameters, and these parameters are uncertain. The LFT therefore explicitly addresses how robust its results are with respect to parameter uncertainty. Each experiment is executed multiple times using combinations of several parameters which are particularly important and uncertain, and the results analysis considers how consistent the importance of factors is among these multiple executions (Section 3).

1.3. Interactions among factors

Managers using the LFT are likely to often be concerned about interactions among factors: would the LFT produce different results and lead to different conclusions if different conditions were assumed in the model inputs? Would it produce different results, for example, if the number of spawners was low instead of high, or for years with high vs. low winter flows?

Designing the LFT to automatically evaluate such interactions would make it substantially more complex and difficult to use and interpret. Therefore, the way to address potential interactions is the simplest: by just executing the LFT for several sets of conditions (low vs. high spawner abundance; low vs. high runoff years) and looking for differences in results.

1.4. When to use the LFT

For its primary purpose of analyzing limiting factors to support management decisions, the LFT should of course be used after *inSALMO* has been fully applied and calibrated to the study site, with the best available input and parameter values. (Even then, LFT results might identify processes and parameters that deserve additional scrutiny because of their importance.)

However, the LFT can be very useful in earlier stages of applying *inSALMO*. Its sensitivity experiments provide information that can be useful, for example, in calibrating the model. The LFT can be thought of as a model exploration tool potentially useful throughout the model application process.

1.5. LFT parameters

The experiments executed by the LFT are controlled by a number of parameters that belong not to *inSALMO* but just to the LFT. The parameters are defined throughout Section 2, summarized with example values in Section 5, and indexed.

2. Limiting Factor Experiments

The following experiments are built into the LFT. Each experiment executes *inSALMO* for each of a specified number of <u>scenarios</u>. Each scenario uses standard inputs except for the one factor being analyzed (and the parameters varied to evaluate parameter uncertainty effects; Section 3). The scenarios vary that factor over a wide range, from below to above the values represented in the standard input. The number of scenarios is set by the LFT parameter *numScenarios*, which must have a value of 2 or higher. Section 7 discusses computational issues that are important in selecting the number of scenarios.

To make results easy to interpret, the LFT analyzes only two outputs from *inSALMO*. First is the total number of live outmigrants from the simulated system, over all years of the model run. Second is the number of "successful" outmigrants: those with length above a threshold that

indicates that they have grown in the simulated system and hence are more likely to survive to the ocean. This threshold is set by the LFT parameter *outmigrantSuccessLength* (cm).

It is very important to understand that the LFT applies the same experiments and parameters to all habitat reaches in an *inSALMO* application. If the application includes multiple reaches differing widely in factors such as flow and temperature, the results may not be as meaningful.

The LFT also applies its changes to all salmon species or races represented in the *inSALMO* application.

2.1. Base flow

This experiment examines flow as a limiting factor. The base flow experiment is designed around the assumption that flows at the study site are controlled by reservoir releases and, possibly, in part by tributary inflow. The experiment simulates the effect of changing the reservoir releases that make up base flow, without changing the tributary inflows and flow variability in the flow input data.

Each scenario adds a constant to the daily flow input, for all simulated days. The constant flow change varies among the scenarios over a range defined by parameters *baseFlowRangeLow* and *baseFlowRangeHigh* (both with units of m^3/s). Any negative flows are set to 0.1 m^3/s.

2.2. Food availability

Production of invertebrate food for juvenile salmon is rarely managed directly, but there are management actions that can indirectly affect food availability. Examples include watershed restoration efforts to control fine sediment inputs and addition of salmon carcasses to increase nutrient availability.

The food availability experiment varies both kinds of food represented in *inSALMO*, assuming that changes in productivity would be reflected in both drift and stationary search food. The standard values of reach habitat variables *habDriftConc* and *habSearchProd* are both multiplied by a ratio that varies among scenarios within a range defined by LFT parameters *foodAvailabilityRatioLow* and *foodAvailabilityRatioHigh*. Food availability can vary widely (over orders of magnitude) among measured values, in part because it is naturally variable and difficult to measure precisely. However, useful values of *inSALMO*'s food availability parameters vary over narrower ranges because they represent long-term and spatial averages. Useful ranges for LFT experiments are *foodAvailabilityRatioLow* = 0.5 and *foodAvailabilityRatioHigh* = 2.0.

2.3. Winter water temperature

Winter water temperature is defined as daily mean water temperature, for days in a winter period. The beginning and ending dates of the "winter" period are defined by the LFT parameters *winterTemperatureStartDay* and *winterTemperatureEndDay*. In California, winter can be defined as November-April, the months when temperatures are generally lowest. For fall Chinook, this period includes almost all the egg incubation period and early fry rearing.

The experiment adds a constant to each day's daily temperature input, for days within the winter period. This constant varies among the scenarios over a range set by the parameters

winterTemperatureRangeLow and *winterTemperatureRangeHigh* (°C). Any negative temperatures are set to zero.

2.4. Summer water temperature

The summer water temperature experiment is identical to the winter temperature experiment, except that temperatures are manipulated for days in warm months. The "summer" period is defined by parameters *summerTemperatureStartDay* and *summerTemperatureEndDay*. (The summer period can overlap with the "winter" period used in the winter water temperature experiment.) In California, May through October generally have the warmest water temperatures. For fall Chinook, this period typically includes the end of the juvenile rearing and outmigration period, and much of the adult arrival and spawning period.

The range of the constant (°C) added to each summer day's temperature input is controlled by the LFT parameters *summerTemperatureRangeLow* and *summerTemperatureRangeHigh*. Any negative temperatures are set to zero.

2.5. Spawning gravel availability

This experiment examines the effect of spawning gravel availability by varying (a) how many cells have spawning gravel, and (b) the amount of cell area containing gravel. The parameters *gravelAvailabilityRangeLow* and *gravelAvailabilityRangeHigh* define the limits of a variable (referred to here as *G*) for the amount of gravel simulated, relative to the amount in the standard input for the modeled sites. The spawning gravel experiment manipulates the variable *cellFracSpawn*, the fraction of cell area that is spawning gravel, for each habitat cell.

For values of *G* less than 1.0, the experiment makes these changes to each cell:

- The standard value of *cellFracSpawn* is multiplied by *G*. Hence, the area of gravel in cells containing gravel is reduced.
- If the standard value of *cellFracSpawn* is greater than zero, its value is set to zero if a random Bernoulli trial is true. The probability of this trial being true is (1-*G*). Therefore, the number of cells containing gravel is reduced.

For values of *G* equal to 1.0, no changes to spawning gravel are made. When *G* is greater than 1.0, these changes are made to each cell:

- The value of *cellFracSpawn* is multiplied by *G*, but limited to a maximum of 1.0. The area of gravel in cells containing gravel is therefore increased.
- If *cellFracSpawn* is zero, its value is set to 0.5 if a random Bernoulli trial is true. The probability of this trial being true is (*G*-1.0). Therefore, the number of cells containing gravel is increased.

The parameters *gravelAvailabilityRangeLow* and *gravelAvailabilityRangeHigh* are limited to values that produce meaningful values of *G*. When *G* is 0.0, there will be no spawning gravel, and when *G* is 2.0 all cells will have gravel. Therefore, the value of *gravelAvailabilityRangeLow* must be between 0.0 and 1.0, and *gravelAvailabilityRangeHigh* must be between 1.0 and 2.0.

2.6. Velocity shelter availability

The velocity shelter experiment examines the value of cover such as large rocks that reduce the swimming speed of drift-feeding fish. It manipulates the values of the cell variable *cellFracShelter*. The experiment works exactly as the spawning gravel experiment does, using parameters *shelterAvailabilityRangeLow* and *shelterAvailabilityRangeHigh* to control the range of velocity shelter availability.

2.7. Hiding cover availability

The hiding cover experiment evaluates the potential benefit of cover such as submerged trees and rocks with crevices, which provide places to hide from predators. It manipulates the cell variable *cellDistToHide*, a characteristic distance (m) between fish feeding in the cell and the nearest hiding cover. The risk from both fish and terrestrial predators (birds, mammals) in *inSALMO* is sensitive to small values of *cellDistToHide* but not to large values: hiding cover is assumed to reduce predation risk only if it is close.

The method for evaluating hiding cover as a limiting factor is similar to that for spawning gravel. The LFT parameters *hidingCoverRangeLow* and *hidingCoverRangeHigh* define the limits of a variable *H* for the simulated availability of close hiding cover in each scenario, relative to the standard input data. “Close” hiding cover is defined by the LFT parameter *hidingCoverThreshold*: hiding cover in a cell is assumed to be valuable to juvenile salmon if *cellDistToHide* is less than *hidingCoverThreshold*. The value of *hidingCoverThreshold* can, for example, be set to a value of 1.0 m.

For values of *H* less than 1.0, the experiment makes these changes to each cell:

- The value of *cellDistToHide* is divided by *H*. Hence, the distance to hiding cover is increased. If *H* is zero, *cellDistToHide* is set to 99 m, a distance at which hiding cover has no benefit.

For values of *H* equal to 1.0, no changes to *cellDistToHide* are made. When *H* is greater than 1.0, these changes are made to each cell:

- The value of *cellDistToHide* is divided by *H*. The distance to hiding cover is therefore reduced in all cells.

- If *cellDistToHide* is greater than *hidingCoverThreshold*, its value is set to (0.5 × *hidingCoverThreshold*) if a random Bernoulli trial is true. The probability of this trial being true is (*H*-1.0). Therefore, the number of cells providing close hiding cover is increased.

The parameters *hidingCoverRangeLow* and *hidingCoverRangeHigh* are limited to produce meaningful values of *H*. When *H* is 0.0, distances to hiding cover will be high for all cells. When *H* is 2.0 all cells will have close hiding cover. Therefore, the value of *hidingCoverRangeLow* must be between 0.0 and 1.0, and *hidingCoverRangeHigh* must be between 1.0 and 2.0.

Users should interpret results of this experiment carefully, as it does not consider the possibility that cover provided to protect juvenile salmon could also shelter the piscivorous fish that prey on them.

2.8. Piscivory risk

This experiment varies the *inSALMO* parameter (*mortFishAqPredMin*) that controls the risk of juvenile salmon being eaten by fish. This parameter is a daily survival probability. It is generally easier to think about predation as a risk (probability of being eaten per day); survival equals one minus risk. The experiment varies the fish predation risk (one minus *mortFishAqPredMin*) over a range.

The range variation in risk, as a fraction of its standard value, is defined by the LFT parameters *piscivoryRiskRatioLow* and *piscivoryRiskRatioHigh*; typical values for these parameters are 0.9 and 1.1. (These risks are *daily* probabilities of dying; survival over a number of days varies sharply with seemingly small changes in daily risk.)

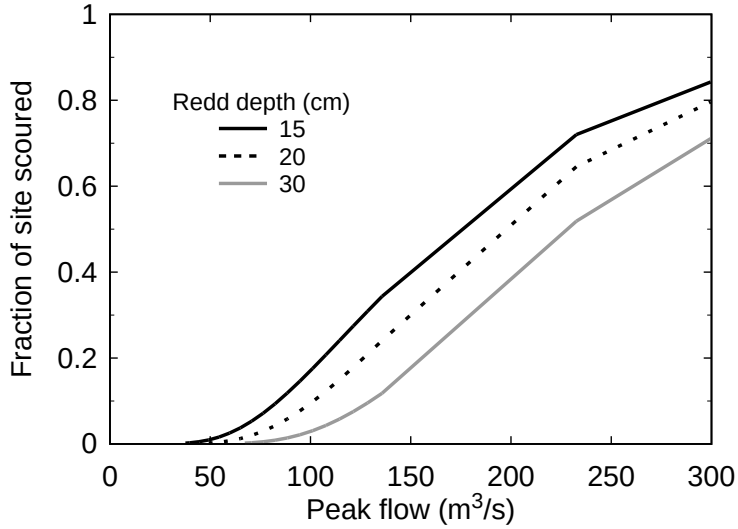
The risk for each scenario is set by multiplying (1.0 - *mortFishAqPredMin*) by the risk ratio. A typical value of *mortFishAqPredMin* is 0.92, corresponding to a daily risk of 0.08. For a range of 0.9 to 1.1, this risk would range among scenarios from 0.072 to 0.088. The values of *mortFishAqPredMin* then range 0.928 to 0.912. The resulting probability of surviving 30 days then ranges 0.11 to 0.063.

(Keep in mind that *mortFishAqPredMin* is the daily survival of juveniles in the riskiest possible habitat; the actual survival of individual salmon is adjusted upwards in accordance with factors such as fish length, depth, and distance to hiding cover.)

Unlike the other experiments, there is no clear and direct relation between the variable representing piscivory as a “limiting factor” and any management variable that could be measured or controlled in the field. While we expect that the value of *mortFishAqPredMin* should increase with the density of predatory fish such as bass and pikeminnow, the relation between piscivore density and simulated predation risk is not clear. The lack of a clear relation is partly because it is not clear how the density of predators affects the individual risk for salmon; doubling the number of predators may not double the risk because, e.g., predators interfere with each other. The ability of model fish to compensate for changes in risk by changing their choice of habitat also makes the relation between piscivore density and salmon survival complex.

2.9. Redd scouring

This experiment evaluates the importance of scouring mortality of redds during high flows, by varying a parameter that controls how vulnerable a redd is to being scoured. The redd scour in *inSALMO* represents the probability of each redd being scoured each time the flow peaks during incubation; this probability is a function of parameters relating reach-scale bed shear stress to flow and a parameter (*mortReddScourDepth*) for how deeply buried redd egg pockets are. The following figure illustrates that, especially at flows where scouring is just initiated, the extent of scouring is strongly affected by *mortReddScourDepth*: scouring is much more extensive and occurs at lower flows when *mortReddScourDepth* is lower.



The redd scour experiment simply varies the value of *mortReddScourDepth*, with scour being more common at low values. The standard value of *mortReddScourDepth* is multiplied by a factor bounded by the parameters *reddScourRatioLow* and *reddScourRatioHigh*.

2.10. Number of spawners

This experiment is designed to evaluate how important the number of spawners is to the number of successful juvenile outmigrants. *inSALMO* includes processes such as redd superimposition and competition among juveniles that can make the relation between number of spawners and number of successful juveniles nonlinear.

The spawner experiment simply multiplies the standard number of spawners (both male and female) by a fraction (ratio) that varies among scenarios. The range in the number of spawners is set by the LFT parameters *spawnerNumberRatioLow* and *spawnerNumberRatioHigh*.

3. Uncertainty Consideration

This section describes how the LFT considers parameter uncertainty in *inSALMO*. Uncertainties in *inSTREAM*, the trout model predecessor to *inSALMO*, have been thoroughly investigated[2] and uncertainty considerations are discussed in Section 15 of the *inSTREAM* documentation[3]. These analyses showed that primary results of these models (e.g., predicted fish abundance) can be quite sensitive to some parameters that have relatively uncertain values, but *relative* results (e.g., the ranks of alternative management scenarios; the relative strength of potential

---

[2] Cunningham, P. M. 2007. A sensitivity analysis of an individual-based trout model. MS. Humboldt State University, Arcata, CA.

[3] Railsback, S. F., B. C. Harvey, S. K. Jackson, and R. H. Lamberson. 2009. InSTREAM: the individual-based stream trout research and environmental assessment model. PSW-GTR-218, USDA Forest Service, Pacific Southwest Research Station, Albany, California.

limiting factors) are typically much less affected by parameter uncertainty. Demonstrating this robustness is important, though, for developing confidence in model results. Therefore, the LFT uses a limited form of parameter uncertainty analysis to evaluate how robust its results are.

Parameter uncertainty analysis usually involves running a model repeatedly using many combinations of values for all the uncertain parameters. This full approach would be computationally infeasible for the LFT because of how many parameters *inSALMO* has and because the limiting factors evaluation by itself requires many model runs.

To make the uncertainty consideration feasible, the LFT allows the user to identify up to three habitat or fish parameters that are varied to represent parameter uncertainty. The user also specifies low and high values for these uncertainty parameters, and the number of values for each parameter. The LFT then executes each scenario of each experiment once for each value of each uncertainty parameter. Hence, if three uncertain parameters are used, and three values specified for each, then each scenario is simulated 27 times. Computational considerations (Section 7) typically limit experiments to three values of one or two uncertainty parameters, so each scenario is executed three or nine times.

The number of uncertainty parameters is set by the LFT parameter *numUncertaintyParams*, which can have values between 0 and 3. The number of values per uncertainty parameter is set by the LFT parameter *numUncertaintyScenarios*; typically, values above 3 for this parameter are computationally infeasible.

The up-to-three uncertainty parameters are specified by LFT parameters *uncertaintyParam1*, *uncertaintyParam2*, and *uncertaintyParam3*. The values of these parameters are just the names of either fish or habitat parameters in *inSALMO*. The value of *uncertaintyParam3* etc. can also be set to “none” when fewer than three uncertainty parameters are used. The low and high values of these uncertainty parameters are calculated by the LFT by multiplying their standard values by the LFT parameters *uncertaintyParam1LowFactor*, *uncertaintyParam1HighFactor*, etc. (The standard value is the parameter value specified by the user as model input.)

The LFT’s default uncertainty consideration uses three *inSALMO* parameters that are particularly important for juvenile fish and relatively uncertain. First is the fish parameter *fishRespParamA*, which relates respiration (metabolic energy cost) to fish weight. Respiration strongly affects the interactions among fish physiology, behavior, survival, and growth (in many individual-based models, not just *inSALMO*). Second is a parameter relating maximum sustainable swimming speed to fish length, *fishMaxSwimParamA*. Maximum swimming speed is another key physiological process affecting both survival and growth. The third parameter is *fishOutmigrateSuccessL1*, which controls the relation between salmon length and tendency to migrate downstream.

The uncertainty in *fishRespParamA* can be assumed ± 20%, so the default values of *uncertaintyParam1LowFactor* and *uncertaintyParam1HighFactor* are 0.8 and 1.2. For *fishMaxSwimParamA*, uncertainty can be assumed ± 20%, so the default values of *uncertaintyParam1LowFactor* and *uncertaintyParam1HighFactor* are 0.8 and 1.2. The uncertainty in *fishOutmigrateSuccessL1* is assumed to be ± 40%, so the default values of *uncertaintyParam2LowFactor* and *uncertaintyParam2HighFactor* are 0.6 and 1.4. To limit the

computational effort, only two values (the low and high end of the ranges) are used for these uncertainty parameters.

This uncertainty formulation means that each experiment will be run for each of eight combinations of parameter values for *fishRespParamA*, *fishMaxSwimParamA*, and *fishOutmigrateSuccessL1*. With 10 potential limiting factors and *numScenarios* set to 5 values, one complete limiting factors analysis therefore requires 400 model runs.

4. Results Analysis and Interpretation

4.1. Analysis methods and interpretation of results

The LFT includes an analysis spreadsheet that applies a standard linear regression approach to evaluate the relative strength of each factor's effect. The basic approach is to regress the model results—total number of outmigrants and number of big outmigrants—against the value of each factor. The independent (x) variable in the regression is not the actual value of the factor being analyzed (food concentration, base flow, etc.) but instead a scaled representation of the factor. The scaled values range from 0.0 for the lowest value of the factor simulated, to 1.0 for the factor's highest value. Other values are scaled between 0.0 and 1.0; hence, experiments with four scenarios for each factor have scaled factor values of 0.0, 0.33, 0.67, and 1.0. The purpose of this scaling is to make results comparable across all factors.

The slope of the regression indicates the strength of the factor's effect on model results: higher slope magnitudes (the absolute value of regression slope) indicate factors with stronger effects on juvenile salmon production. The squared correlation coefficient R^2 indicates how consistent or robust the factor's effect is over its range and among the uncertainty parameter values. Results of LFT experiments generally fall within three categories that can be interpreted as follows:

- Relatively high slope magnitude and high R^2: the factor has a strong and consistent effect on juvenile salmon production.

- Relatively high slope magnitude but low R^2: the factor has a strong effect on salmon production but the effect is nonlinear or interacts with the parameters varied to represent uncertainty (Section 3).

- Low slope magnitude and low R^2: the factor has relatively little effect on salmon production. <u>However, it is possible instead that the factor has strong but highly nonlinear effects</u>.

4.2. Results formats

The LFT's analysis spreadsheet presents results in two formats that are both essential to review. The first format is a table of the statistical results (slope magnitude and R^2) for each factor and each of the two kinds of results (total and big outmigrant numbers). This table is useful for quick and reproducible comparison of the different factors.

The second results format is the actual plots used to produce the regression statistics. Examination of these plots is important for understanding each factor's effect, especially to look

for nonlinear responses. The plots are especially important for factors falling in the third category listed in Section 4.1: factors that are indicated by the statistics to have little effect but may in fact have strong but highly nonlinear effects. If, for example, the summer temperature experiment produces a low slope and R^2, does that mean summer temperature has little effect or that both decreasing and increasing the temperature from the standard input have strong but negative effects? This question is easily answered by looking at the results plot.

5. Parameters Controlling the Limiting Factors Tool

The LFT parameters are listed in Table 1. The table also includes example values, which are recommended for the Clear Creek, California, study site to which the tool is first applied. Values should be reconsidered for other sites, considering the following guidance.

The range over which management factors are varied in each experiment includes values both below and above the standard values specified by the input to *inSALMO* for a study site. These ranges should approximate the range of feasible management actions, and not be broader than realistic. Using overly broad ranges of experimental factors increases the possibility of nonlinear responses that cause misinterpretation of limiting factors results, or results that are unhelpful because they represent conditions that cannot be attained by feasible management actions.

The range of variation need not be symmetric around the standard input. For factors such as spawning gravel availability that are much more likely to be increased instead of decreased by management actions, the range evaluated in the limiting factor experiments should be skewed toward higher values.

Table 1. Parameter definitions and example values for the LFT.

Parameter	Meaning	Typical value for Clear Creek
numScenarios	The number of senarios per experiment (applies to all experiments)	5 (see Section 7)
outmigrantSuccessLength	The length (cm) above which outmigrants are considered “successful” in results analyses	5.0
winterTemperatureStartDay	The first day (MM/DD) of the “winter” period for winter temperature experiments	11/1
winterTemperatureEndDay	The last day (MM/DD) of the “winter” period for winter temperature experiments	3/31
winterTemperatureRangeLow	The constant change in temperature (°C) for the lowest-temperature scenario in the winter temperature experiment.	-4.0

Parameter	Meaning	Typical value for Clear Creek
winterTemperatureRangeHigh	The constant change in temperature (°C) for the highest-temperature scenario in the winter temperature experiment.	4.0
summerTemperatureStartDay	The first day (MM/DD) of the “summer” period for summer temperature experiments	4/1
summerTemperatureEndDay	The last day (MM/DD) of the “summer” period for summer temperature experiments	10/31
summerTemperatureRangeLow	The constant change in temperature (°C) for the lowest-temperature scenario in the summer temperature experiment.	-4.0
summerTemperatureRangeHigh	The constant change in temperature (°C) for the highest-temperature scenario in the summer temperature experiment.	4.0
baseFlowRangeLow	The constant change in flow (m^3/s) for the lowest-flow scenario in the base flow experiment.	-2.0
baseFlowRangeHigh	The constant change in flow (m^3/s) for the highest-flow scenario in the base flow experiment.	4.0
gravelAvailabilityRangeLow	The lowest value of relative gravel availability in the spawning gravel experiment. The value must be between 0.0 and 1.0	0.5
gravelAvailabilityRangeHigh	The highest value of relative gravel availability in the spawning gravel experiment. The value must be between 1.0 and 2.0	1.5
shelterAvailabilityRangeLow	The lowest value of relative velocity shelter availability in the velocity shelter experiment. The value must be between 0.0 and 1.0	0.5
shelterAvailabilityRangeHigh	The highest value of relative velocity shelter availability in the velocity shelter experiment. The value must be between 1.0 and 2.0	1.5
hidingCoverRangeLow	The lowest value of hiding cover availability in the hiding cover experiment. The value must be between 0.0 and 1.0	0.5

Parameter	Meaning	Typical value for Clear Creek
hidingCoverRangeHigh	The highest value of hiding cover availability in the hiding cover experiment. The value must be between 1.0 and 2.0	1.5
hidingCoverThreshold	In the hiding cover experiment, the cell distance to hiding cover (m) defining cover close enough to provide substantial benefit	1.0
piscivoryRiskRatioLow	The change in piscivory risk (fraction of the standard value) for the lowest-risk scenario in the piscivory risk experiment.	0.9
piscivoryRiskRatioHigh	The change in piscivory risk (fraction of the standard value) for the lowest-risk scenario in the piscivory risk experiment.	1.1
foodAvailabilityRatioLow	The fraction of standard food availability for the lowest-food scenario in the food availability experiment.	0.5
foodAvailabilityRatioHigh	The fraction of standard food availability for the highest-food scenario in the food availability experiment.	2.0
reddScourRatioLow	The lowest factor by which redd depth is multiplied in the redd scour experiment	0.5
reddScourRatioHigh	The highest factor by which redd depth is multiplied in the redd scour experiment	1.5
spawnerNumberRatioLow	The fraction by which the standard number of spawners is multiplied in the lowest-number scenario in the spawner number experiment.	0.5
spawnerNumberRatioHigh	The fraction by which the standard number of spawners is multiplied in the highest scenario in the spawner number experiment.	1.5
numUncertaintyParameters	The number of parameters varied to represent parameter uncertainty (Section 3).	3
numUncertaintyScenarios	The number of values used per uncertainty parameter.	2

Parameter	Meaning	Typical value for Clear Creek
uncertaintyParam1	The name of the first uncertainty parameter (a text string with the name of a fish or habitat parameter; set to “none” if no uncertainty parameters are used).	*fishRespParamA*
uncertaintyParam1LowFactor	The ratio multiplied by the standard value of uncertainty parameter 1 to set the low end of its range.	0.8
uncertaintyParam1HighFactor	The ratio multiplied by the standard value of uncertainty parameter 1 to set the high end of its range.	1.2
uncertaintyParam2	The name of the second uncertainty parameter (“none” if fewer than two uncertainty parameters are used).	*fishMaxSwimParamA*
uncertaintyParam2LowFactor	The ratio multiplied by the standard value of uncertainty parameter 2 to set the low end of its range.	0.8
uncertaintyParam2HighFactor	The ratio multiplied by the standard value of uncertainty parameter 2 to set the high end of its range.	1.2
uncertaintyParam3	The name of the third uncertainty parameter.	*fishOutmigrate SuccessL1*
uncertaintyParam3LowFactor	The ratio multiplied by the standard value of uncertainty parameter 3 to set the low end of its range.	0.6
uncertaintyParam3HighFactor	The ratio multiplied by the standard value of uncertainty parameter 3 to set the high end of its range.	1.2

6. Using the Tool

The LFT exists as an option inside *inSALMO*'s graphical user interface (GUI), which is documented separately. When a model project is ready for use (all input files have been provided; parameter values are at least tentatively established; test runs have been completed), the user can hit the "Limiting Factors" button on the GUI interface. (It is generally smart to first save the project as a new project just for the LFT analysis.)

The "Limiting Factors" button takes the user to the GUI's LFT view, which has two tabs labeled "Limiting Factors Tool Parameters" and "Execution Control/Output". Normal use of the LFT follows these steps:

1. On the "Limiting Factors Tool Parameters" tab, edit the LFT parameter values to customize the analysis. Use the information in sections 2, 3, and 5 to select parameter values.

2. Click the "Start Limiting Factors Experiments" button. This creates separate subdirectories (under the project directory) for each experiment and builds the input and setup files (especially Experiment.Setup) for each. It also starts execution of *inSALMO* for each experiment (explained in Section 7).

3. Switch to the "Execution Control/Output" tab (Figure 2) to observe execution of the experiments. Individual experiments can be terminated via a button on their tab. All runs can be terminated at once via a button on the "Limiting Factors Tool Parameters" tab.

4. When all the experiments have stopped (because they finished, you terminated them, or they stopped due to an error), a dialog pops up asking you whether you want to process and view results now or later. If you prefer, you can later re-open the project and LFT, and click on the "Process/View Results" button of the "Limiting Factors Tool Parameters" tab to open the results analysis spreadsheet.

5. (If not all experiments completely finished, you can still use the following steps to observe the partial results.)

6. When you do process/view the results, the LFT opens an Excel spreadsheet that contains code to import LFT results and report the statistical and graphical results. (These results are imported from a special output file named LFT_Output.rpt.) After the spreadsheet has updated itself, you will be asked to save it under a new file name (Figure 3). **This Excel import process will very likely halt, with Excel displaying a "Security warning" saying that macros have been disabled. You must choose the option to allow macro content for the analysis to proceed.**

7. Examine the results displayed in the spreadsheet. It includes a table of statistical results (the "ResultsSummary" sheet) and a separate sheet displaying graphical results for each experiment. The table of statistical results includes a column labeled "n" that reports the number of model runs completed for each limiting factor experiment.

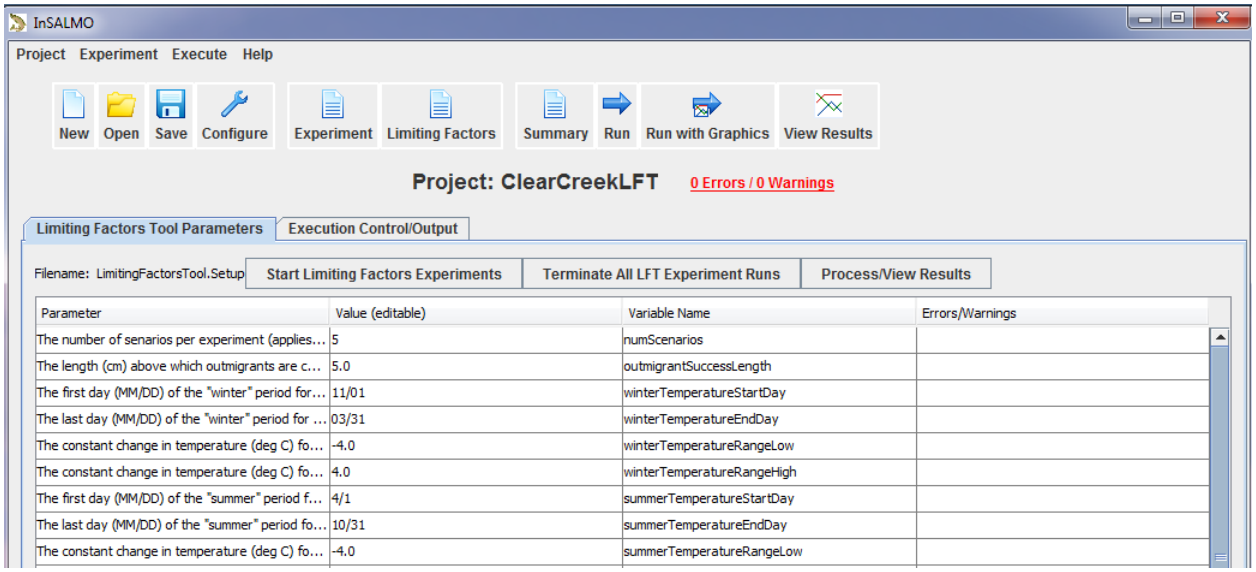


Figure 1. The *inSALMO* GUI's LFT view and its "Limiting Factors Tool Parameters" tab.

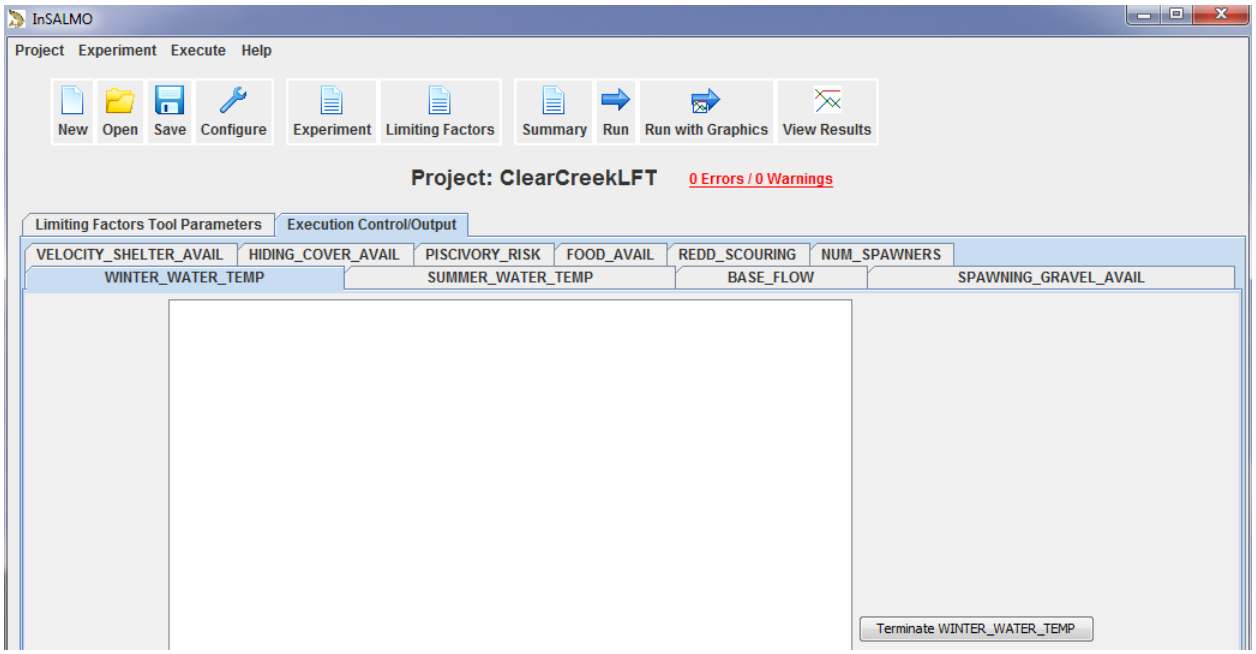
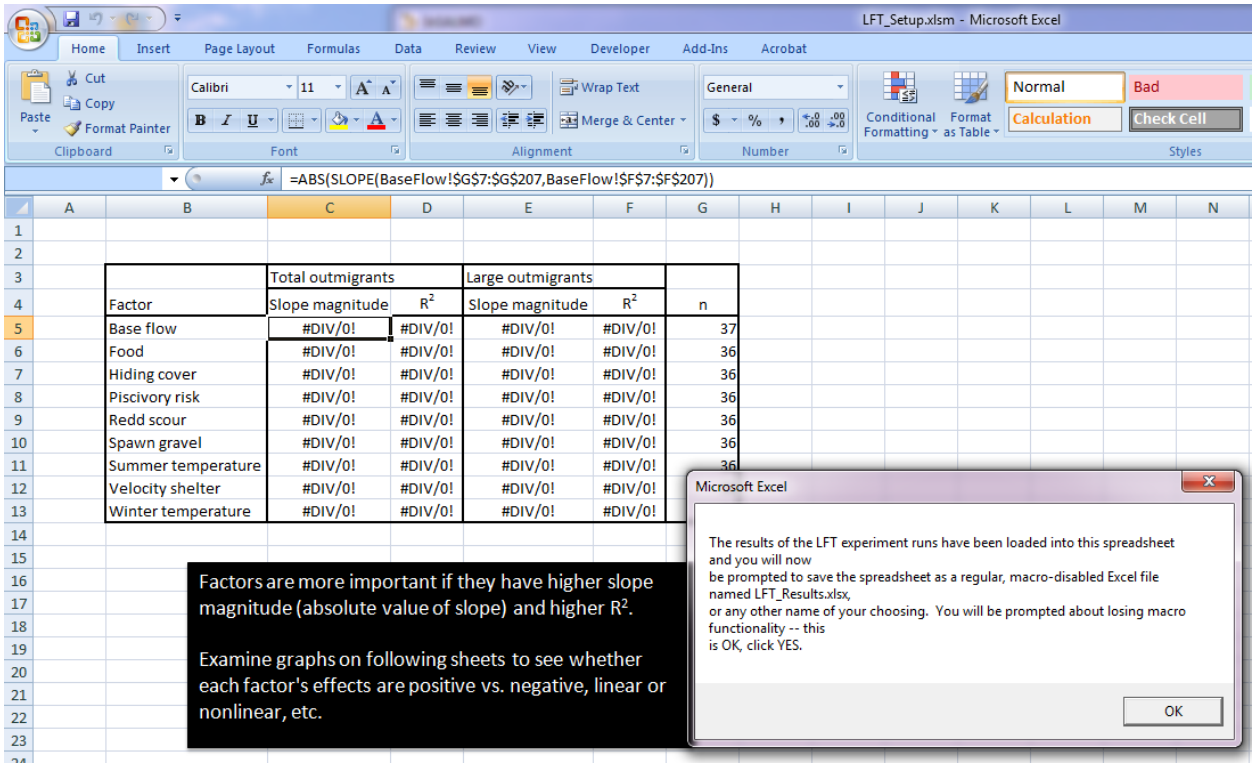


Figure 2. The LFT view and its "Execution Control/Output" tab. A separate tab is created for each LFT experiment, with an output window where *inSALMO* output is displayed as the model runs.



Factor	Total outmigrants Slope magnitude	Total outmigrants R^2	Large outmigrants Slope magnitude	Large outmigrants R^2	n
Base flow	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	37
Food	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	36
Hiding cover	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	36
Piscivory risk	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	36
Redd scour	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	36
Spawn gravel	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	36
Summer temperature	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	36
Velocity shelter	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	
Winter temperature	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	

Figure 3. The LFT's analysis spreadsheet automatically imports results and asks you to save the file as a regular Excel file.

7. Computational Considerations

The limiting factors analyses are computationally demanding and the extent of the analysis is typically limited by computer considerations. The computational limitations are not just due execution speed (which can be overcome with patience or a faster computer) but can also be due to memory availability.

When the user starts the limiting factor experiments, the LFT starts each experiment (as defined in Section 2) as a separate *inSALMO* job: a set of model runs that are executed sequentially. All the experiment jobs are started at once and the operating system automatically assigns them to the available processors. The experiments therefore are executed more or less in parallel, with execution time depending on how many processors the computer has and how fast they are.

On a modern computer (e.g., Intel Core chip sets), the number of experiment jobs seems to affect only execution time: the jobs simply rotate to the available processor cores. (The LFT should not be used on old computers with Pentium-type hardware.) The *inSALMO* code is memory-intensive, but execution does not appear to slow substantially when the available RAM is filled (the processor moves inactive memory use to the hard disk's swap file; however, a job may be killed by the operating system when the swap file is full).

Experience to date indicates that the most likely computation problem is that extremely large experiment jobs stop before completing because the model uses up all the available memory. The number of runs per LFT analysis is equal to 10 (the number of factors analyzed) × *numScenarios* × U^n where U is the number of values per uncertainty parameter (LFT parameter *numUncertaintyScenarios*) and *n* is the number (between 0 and 3) of uncertainty parameters used (LFT parameter *numUncertaintyParams*; Section 3). On a Windows computer with relatively high random-access memory (RAM; e.g., 8 gigabytes), LFT analyses with 1000+ model runs should be feasible.

(Large analyses can also fail because of a Windows limitation on how much memory can be allocated to each experiment and fragmentation of that memory. The symptom of this problem is that the model stops with an error in the function "malloc". If very large experiments stop due to inadequate RAM or other memory problems, the only solution for Windows users is to set up smaller experiments by using fewer scenarios, uncertainty parameters, or values per uncertainty parameter. Another solution to this memory allocation limitation is to use the LFT in Windows to set up the limiting factor experiments and then execute them in a 64-bit Linux installation of *inSALMO*, which is much less subject to memory limitations. The LFT can be started in Windows and used to set all the LFT parameter values. The "Start Limiting Factor Experiments" button is then clicked, after which the "Terminate All LFT Experiment Runs" button is clicked to stop the runs. This will create all the separate directories and input files for each experiment; these can be copied to Linux for execution there. An especially large swap file in the Linux installation can offset the graduate consumption of memory by *inSALMO*.)

LFT Parameter Index

B

baseFlowRangeHigh 8, 16
baseFlowRangeLow 8, 16

F

foodAvailabilityRatioHigh 8, 17
foodAvailabilityRatioLow 8, 17

G

gravelAvailabilityRangeHigh 9, 16
gravelAvailabilityRangeLow 9, 16

H

hidingCoverRangeHigh 10, 17
hidingCoverRangeLow 10, 16
hidingCoverThreshold 10, 17

L

LFT_Output.rpt 19

N

numScenarios 7, 15, 22
numUncertaintyParameters 17
numUncertaintyParams 13, 22
numUncertaintyScenarios 13, 17, 22

O

outmigrantSuccessLength 8, 15

P

piscivoryRiskRatioHigh 11, 17
piscivoryRiskRatioLow 11, 17

R

reddScourRatioHigh 12, 17
reddScourRatioLow 12, 17

S

shelterAvailabilityRangeHigh 10, 16
shelterAvailabilityRangeLow 10, 16
spawnerNumberRatioHigh 12, 17
spawnerNumberRatioLow 12, 17
summerTemperatureEndDay 9, 16
summerTemperatureRangeHigh 9, 16
summerTemperatureRangeLow 9, 16
summerTemperatureStartDay 9, 16

U

uncertaintyParam1 .....13, 18
uncertaintyParam1HighFactor 13, 18
uncertaintyParam1LowFactor 13, 18
uncertaintyParam2 .....13, 18
uncertaintyParam2HighFactor 13, 18
uncertaintyParam2LowFactor 13, 18
uncertaintyParam3 .....13, 18
uncertaintyParam3HighFactor 18
uncertaintyParam3LowFactor 18

W

winterTemperatureEndDay 8, 15
winterTemperatureRangeHigh 9, 16
winterTemperatureRangeLow 9, 15
winterTemperatureStartDay 8, 15