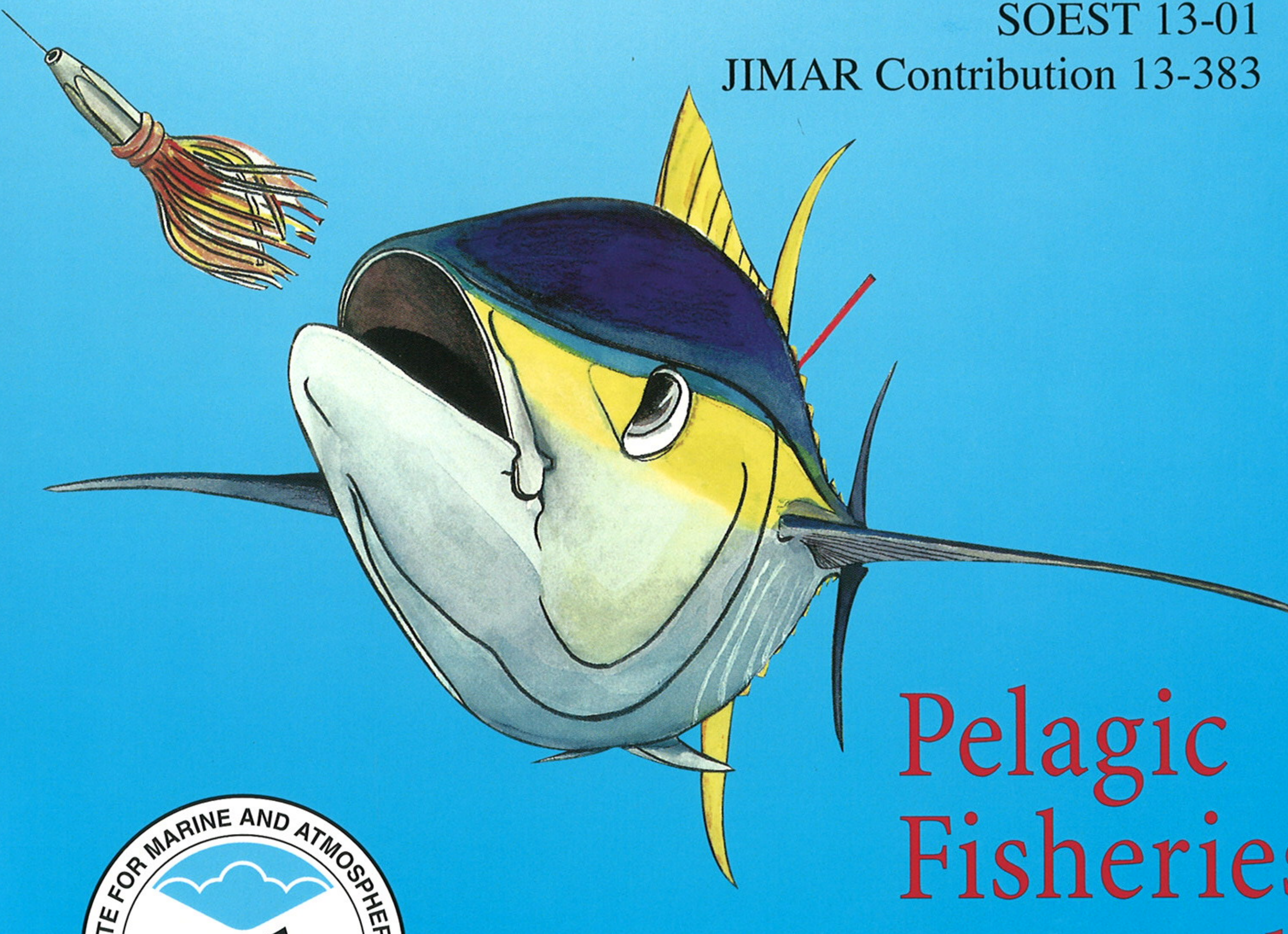
Agent-Based Fishery Management Model of Hawaii's Longline Fisheries (FMMHLF): Model Description and Software Guide

Run Yu, Steve Railsback, Colin Sheppard, and PingSun Leung

SOEST 13-01
JIMAR Contribution 13-383



Pelagic Fisheries Research Program

AGENT-BASED FISHERY MANAGEMENT MODEL OF HAWAII'S LONGLINE FISHERIES (FMMHLF): MODEL DESCRIPTION AND SOFTWARE GUIDE

Run Yu
Dept. of Natural Resources and
Environmental Management
University of Hawaii at Manoa,
Honolulu, HI 96822

Steve Railsback
Lang Railsback & Associates and
Humboldt State University
Arcata, CA 95521

Colin Sheppard
Arcata, CA 95521

PingSun Leung
Dept. of Natural Resources and
Environmental Management
University of Hawaii at Manoa,
Honolulu, HI 96822

SOEST 13-01
JIMAR Contribution 13-383

AGENT-BASED FISHERY MANAGEMENT MODEL OF HAWAII'S LONGLINE FISHERIES (FMMHLF):

A. Model Description

Table of Contents

1 **INTRODUCTION** 1
2 **OVERVIEW** 1
2.1 Purpose 1
2.2 Entities, State Variables, and Scales 2
2.2.1 Fishery 2
2.2.2 Grid cells 4
2.2.3 Fishing vessels 5
2.2.4 Social networks 7
2.3 Process Overview and Scheduling 7
2.3.1 HLF actions 8
2.3.2 Grid cell actions 8
2.3.3 Fishing vessel actions 9
2.3.4 Output 12
3 **DESIGN CONCEPTS** 13
3.1 Basic Principles 13
3.2 Emergence 13
3.3 Adaptation 13
3.4 Objectives 13
3.5 Learning 14
3.6 Prediction 14
3.7 Sensing 14
3.8 Interaction 15
3.9 Stochasticity 15
3.10 Collectives 15
3.11 Observation 15
4 **DETAILS** 15
4.1 Initialization 15
4.1.1 HLF 16
4.1.2 Cells 16
4.1.3 Vessels 16
4.1.4 Social networks 16
4.2 Input data 17
4.2.1 Fish prices 17
4.3 Submodels 17
4.3.1 Cell catch and turtle interaction rate updates 17
4.3.2 Fishing trip decision 18
4.3.3 Respond to closure 22
4.3.4 Fishing cell selection 23

4.3.5 Fish catch 24
4.3.6 Turtle interaction....24
4.3.7 Return to port decision....24
4.3.8 Fish sale 27
APPENDIX A. DEVELOPMENT OF PARAMETER VALUES OF HLF 28

Figures

1. The spatial extent and ocean regions of the Hawaii application of FMMHLF. 4
2. Overview of fishing vessel states and decisions. 9
3. Contours of Interaction-weighted Net Daily Return (INDR) vs. vessel's CPUE and price for SWF, assuming no probability of fishery closure due to turtle interactions. 21
4. Interaction-weighted Net Daily Return (INDR) vs. number of turtle interactions, out of a quota of 21 interactions, at June 1. Three SWF prices are used. 22
5. Interaction-weighted Net Daily Return (INDR) vs. number of turtle interactions, out of a quota of 21, for four trip departure dates. 22
6. Exploration of the return to port submodel: plots of how expected net revenue for an additional day of fishing varies with (X) the estimated daily catch made in trip planning and (Y) the average catch experienced during the fishing trip. Assumptions include daily operating cost = $1000 and fish price of $2.36/lb. Left panel: expected net revenue after 1 day of fishing, which depends equally on the expected and experienced catch. Right panel: expected net revenue after 4 days of fishing, which varies much more with experienced catch. 26

Tables

1. State variables of the HLF. 3
2. State variables of cells. 5
3. State variables of vessels. 6
4. Fishing set variables. 7
5. Initialization of HLF variables. 16

Terminology Index

bigeye tuna (BYT) 2
by-catch 2
catch per unit effort (CPUE) 4
catch quota 3
catch species 2
date closure 3
effort quota 3
fishing set 5
gear type 2
Hawaii's longline fisheries (HLF) 1
ocean regions 2
other fish (OTHER) 2
swordfish (SWF) 2
target species 2
tuna (TUNA) 2
turtle interaction 1
turtle interactions per unit of effort (IPUE) 4
year type 3
yellowfin tuna (YFT) 2

1 INTRODUCTION

This document describes version 3.0 of the Fishery Management Model of Hawaii's Longline Fisheries (FMMHLF), which is being developed by the Department of Natural Resources and Environmental Management of the University of Hawaii under the NSF grant "An exploratory application of agent-based modeling for policy evaluations in Hawaii's longline fishery" (No. SES-0918185). This document follows the "Overview, Design Concepts, Details" (ODD) standard protocol for describing agent-based models (Grimm et al. 2010. The ODD protocol: a review and first update. *Ecological Modelling* 221:2760-2768). The Overview (Section 2) provides a broad summary of the model, beginning with its purpose then moving on to its overall structure. Section 3 describes the model's conceptual framework, and Section 4 provides full detail on the model's methods and inputs.

Although this model was developed for Hawaii's longline tuna and swordfish fisheries, it is designed to be flexible so that the modeling framework could potentially be adapted for other longline fisheries. Parameter values and inputs for the Hawaii fishery are used in this document as examples. Please see Appendix A for a brief description of the development of the parameters and the input data of Hawaii longline fisheries. However, details of the development process and input data are documented separately.

2 OVERVIEW

2.1 Purpose

The FMMHLF is an agent-based simulation model designed for assessing the potential impacts of alternative fisheries regulatory policies on Hawaii's longline fisheries (HLF). By HLF we refer to Hawaii's fleet of longline fishing vessels and their owners and operators; and the regulatory agencies and markets the fleet interacts with. The model is specifically designed to represent the State's largest fleet, which uses Honolulu as its homeport.

The primary regulatory policies of interest in FMMHLF, or concern to both fishers and regulatory agencies, are those that protect sea turtles. Sea turtles can be disturbed or harmed when they become entangled or hooked on fishing lines; an event termed "turtle interaction". Currently, turtles are protected by an annual quota (or "cap") on turtle interactions. Under this policy, if the number of turtle interactions in the current calendar year reaches the cap, then longline swordfish fishing is prohibited until the end of the year.

The intent of the FMMHLF is to capture the key elements that influence fishing decisions of individual vessels that make up HLF and thus predict and assess the possible responses of HLF to regulatory policies. Policy assessment focuses on four aspects of HLF: the allocation of fishing effort between tuna and swordfish fisheries; the spatiotemporal distribution of fishing effort; total catch of the two fisheries; and interaction with protected sea turtles. Additionally, FMMHLF was designed to test how alternative decision rules (e.g., profit-maximizing versus revenue-targeting by vessel operators) and social networks affect the performance of HLF.

2.2 Entities, State Variables, and Scales

This section describes what kinds of things (entities) are in the model, what their key state variables (that describe how important characteristics vary among entities or over time) are, and the scales at which the model operates. FMMHLF simulates four kinds of entities: the overall fishery (including regulators, the market that sets prices, etc.); the grid cells that represent the ocean environment and its fish populations; the individual fishing vessels and operators; and social networks, which are groups of vessels that share information.

In terms of temporal scale, the model uses a one-day time step and simulations can run for many years. However, FMMHLF currently uses input (defining fish catch rates, market prices, etc.) for only one year and then uses that input for all simulated years. Typical simulations run for 20 years, with the first 10 years used to eliminate the effects of initial conditions. Dates are simulated using a simplified calendar where all years are assumed to have 12 months of 30 days; so only 360 days are simulated per year. The spatial scale is discussed in Section 2.2.2.

2.2.1 Fishery

In FMMHLF, the HLF is an entity that represents the regulators and rules, markets, and traditions within which individual vessels operate, and collects data on the fleet of vessels. The key state variables of the HLF are listed in Table 1.

Regulators set management policies such as catch limits, closures (specific areas that are closed to particular fisheries for specific time periods), and rules for protecting sea turtles. In this version of the model, regulators are represented not as agents with behaviors that change regulations during a simulation but as model rules and parameters that define how the fisheries are regulated. These rules and parameters are static; they do not change during a simulation.

The market sets the prices that vessels receive for their catch. This version represents the market simply as a time series of daily prices that are input (Section 4.2).

Some HLF variables have different values for different gear types and catch species. "Gear type" refers to the kind of fish that vessels are rigged for before leaving port, with each fishing trip using one gear type. (The term "fishery" is often used similarly; we use "gear type" to avoid confusion with the other meanings of "fishery".) Gear types are named for their "target species", which refers to the (one or several) species the gear is designed to catch. In this version of FMMHLF, gear types are either swordfish (SWF) or tuna (TUNA). "Catch species" refers to the fish species actually caught, which can differ from the target species because of by-catch (catching species other than those targeted) or, in the case of tuna, because there are several species targeted with the same gear type. Catch species currently in the model are swordfish (SWF), bigeye tuna (BYT), yellowfin tuna (YFT), or other fish (OTHER). All catch species can potentially be caught by any gear type and the parameters control the relation between gear type and the catch of each species.

Variables for harvest regulation have different values for several different regions or fishery management areas (Figure 1); these are referred to as "ocean regions". Quotas are described throughout this model description as annual, which means they are reset at the start of each simulated calendar year in the model. However, this quota reset interval is actually controlled by parameters that define the number of months between resets of each quota; these parameters all have default values of 12 months.

Table 1. State variables of the HLF.

Category	Variable	Description
Fishery status	date	The current date under the model scenario.
	numberOfVessels[1]	Number of vessels that actively fished in the simulated year.
	numberOfTrips[1]	Total number of fishing trips made in the simulated year.
	numberOfSets[1]	Total number of fishing sets made in the simulated year.
	catchSoFar[2]	Total weight (lb) of fish landed by all vessels in a year.
	catchValue[2]	Total value ($) of fish caught and sold in the market in a year.
Turtle interaction regulation	turtleCap[3]	The quota on annual sea turtle interactions (a static parameter). No more use of gear type SWF is allowed in the current calendar year when this quota is reached.
	turtleInteractionsSoFar[3]	The number of turtle interactions in the current calendar year.
Fish harvest regulation	catchQuota[4]	An annual quota on fish catch (total catch of the catch species, lbs); no more fishing with the gear type targeting the species is allowed in the current calendar year when this quota is met. All catch of that species counts toward the quota, even fish caught as by-catch by vessels using gear targeting other species.
	effortLimit[5]	An annual quota on fishing effort (number of sets); no more fishing is allowed, in *all* ocean regions, for the specified gear type in the current calendar year when this limit is met.
	dateClosures	A list of "closures", which are complex variables describing dates and locations at which fisheries are closed. Each closure includes a gear type, the month and day on which the closure begins, the month and day on which it ends, and the coordinates (latitude, longitude) of a polygon. Fishing with the gear type is prohibited inside the polygon on days between the starting and ending days, in all years.
	fisheryIsClosed[1]	Whether the fishery of the specified gear type is closed, in the specified ocean region, for the rest of the calendar year (true/false). This variable only reflects closures due to quotas on catch species, fishing effort, and turtle interactions. Date closures are tracked separately.
Market	priceToday[6]	The market price ($/lb) for fish sold on the current day.
Year type	baseYear	An integer (0 to 5) identifying which one of six year types to simulate. Year types allow some model variables (CPUE; Section 2.2.2) to be given different values to represent different historical conditions.
Global vessel parameters	restingTimeN[5], restingTimeP[5]	Parameters for the negative binomial distribution for resting times between trips (Section 4.3.2).
	maxStorageDays[5]	The maximum number of days fish can be stored without losing their market value (Section 4.3.4).

[1]There are actually separate variables for each gear type and ocean region.
[2]There are actually separate variables for each catch species and each ocean region.
[3]There are separate variables for each turtle species.
[4]There are separate variables for each catch species and ocean region.
[5]There are separate variables for each gear type.
[6]There are separate variables for each catch species.

In addition to catching various fish species, longline fleets could incidentally catch other sea animals, such as sea turtles, in the fishing gears. Loggerhead turtles and leatherback turtles in the North Pacific are classified as endangered species under the Endangered Species Act. To protect these two endangered turtle species, the HLF is operated under strict limits on the number of turtle interactions allowed.

2.2.2 Grid cells

Grid cells are the spatial entities that make up the fishing grounds. The cell size (spatial resolution) is 1° longitude × 1° latitude. In the case of the Hawaiian longline fishery, the spatial extent is a 45×45 grid of cells, from the equator (latitude 0° N) to 45° N and from longitude 135° W to 180° W. Each cell is within an ocean region, of which there are five in the Hawaii application (Figure 1).

The cells are centered on the intersection of lines of latitude and longitude, so the cell at 29° N, 168° W includes the area from 28° 30' to 29° 30' N between 167° 30' and 168° 30' W. For distance and location calculations, the model assumes a Mercator projection and treats each cell as a rectangle 53 miles across east-west and 69 miles in extent north-south. (Distances and speeds are in units of statute, not nautical, miles.) Hence, the distance D (statute miles) between two cells or points is calculated as:

$$D = \sqrt{[69(Lat_1 - Lat_2)]^2 + [53(Lon_1 - Lon_2)]^2}$$

where Lat_1, Lon_1, etc. represent the latitude and longitude of the two points.

The primary purpose of the cells is to represent how potential fish catch and turtle interactions vary over space and time. Fish catch per unit of effort (CPUE) and turtle interactions per unit of effort (IPUE) are the two key state variables (Table 2). CPUE is an indirect indicator of the abundance of fish stock in the fishing cell while IPUE is an indirect indicator of the turtle population in the cell. Changes in CPUE and IPUE over time reflect the seasonal dynamics of fish and turtle stocks.

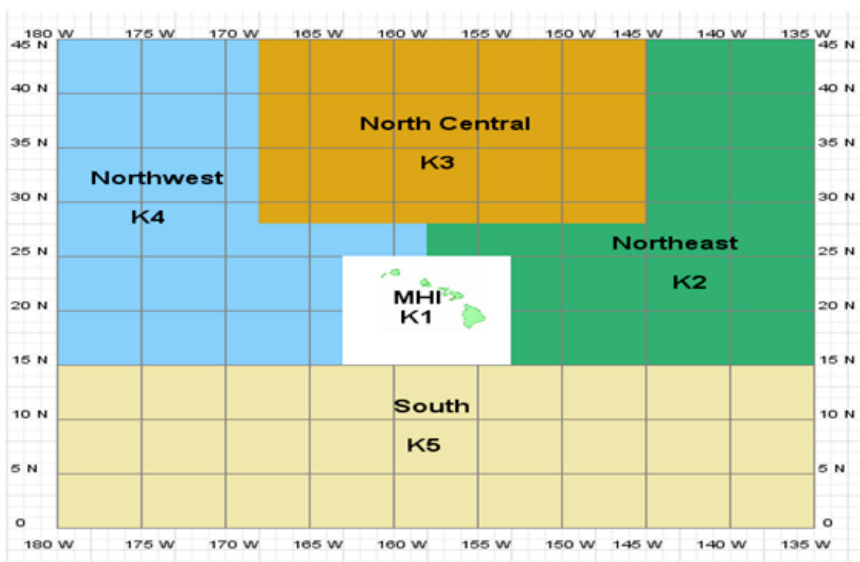


Figure 1. The spatial extent and ocean regions of the Hawaii application of FMMHLF.

Table 2. State variables of cells.

Category	Variable	Description
Cell location (static)	longitude	Longitude of the cell midpoint.
	latitude	Latitude of the cell midpoint.
	oceanRegion	The ocean region (Figure 1) where the cell is located.
Fish catch	CPUEMonthly[1]	Current monthly mean fish catch per unit effort (lbs fish per 1000 hooks). (All cells have the same value of this variable, so in the software it belongs to the HLF.)
	CPUEDaily[1]	Current daily mean fish catch per unit effort (lbs fish per 1000 hooks). (The actual catch of a vessel depends on its characteristics as well as this variable.)
	fisheryIsClosedByDateClosure[2]	Indicates (true/false) whether the cell is currently closed for fishing with a particular gear type, due to a date closure. (Other types of closures are tracked by the HLF.)
Turtle interactions	IPUEMonthly[3]	Current monthly mean probability of turtle interactions per unit of effort (interactions per 1000 hooks in a fishing set). Because this rate is small it is also treated as the probability, per 1000 hooks, of interacting with one turtle on one set. (The actual interaction probability for each vessel depends on vessel characteristics as well as this variable.)
	IPUEDaily[3]	Current daily mean rate of turtle interactions per unit of effort (interactions per fishing set). (In this version IPUEDaily is equal to IPUEMonthly.)

[1]There are separate variables for each gear type and catch species.
[2]Separate variables for each gear type.
[3]Separate variables for each turtle species and gear type.

2.2.3 Fishing vessels

Vessels are agents that represent both fishing boats and the owners and captains operating them. Hence, their state variables (Table 3) include physical characteristics of boats and factors affecting fishing decisions.

The most important behaviors and actions of vessels involve “fishing sets”. A fishing set is the fundamental unit of fishing effort: the deployment and retrieval by a boat of its longline, and resulting fish catch. In FMMHLF, a fishing set is a compound variable that contains information about one such set (Table 4).

Table 3. State variables of vessels.

Category	Variable	Description
Physical characteristics (static)	size	Vessel's length category (small, medium, or large).
	vesselType	The vessel's gear types: either TUNA (the vessel only fishes for tuna) or BOTH (the vessel has gear for TUNA and SWF).
	fuelCapacity	Fuel holding capacity (lbs).
	fuelPerDay	Fuel use when traveling or fishing (lbs/day).
	numberOfCrew	Number of fishermen (other than the captain) on the vessel (not currently used).
	hooks[1]	Number of hooks (1000s) used in a fishing set.
	CPUEAdjFactor[2]	A vessel-specific coefficient for calculation of catch.
	IPUEAdjFactor[3]	Vessel-specific multiplier for probability of interacting with sea turtles.
	speed	Speed when traveling to and from fishing grounds (statute miles/day).
	fishHoldingCapacity	Fish holding capacity (lbs).
	typicalTripLength[1]	Typical or average trip length (number of days at sea), used in planning fishing trips.
	maxDaysInSea	The maximum length (days) of a fishing trip.
Social characteristics (static)	network	The social network (Section 2.2.4) to which the vessel belongs (represented as an integer). The vessel uses the social network's memory of fishing cell characteristics.
	ethnicity	The vessel owner's ethnicity (e.g., Vietnamese, Korean, white). (Not currently used.)
	ownerOperated	True/false for whether the vessel is operated by its owner (not currently used).
Financial characteristics (static)	fixedCost	Fixed cost of entering the fishery ($/year; not currently used).
	operatingCost[1]	Cost of making one fishing set.
	wage	Crew wages ($/day for the entire crew).
	travelCost	Operating cost when traveling to and from fishing grounds ($/day).
	planningHorizon	How far (days) in the future the vessel operator considers when choosing departure dates.
	priceExpLength	The time window (days) used when forecasting fish price.
	priceAdjFactor[2]	The vessel's fish price premium, relative to market price.
	revenueShare	Percentage of net revenue paid to the crew. (Not currently used.)
	revenueTarget	The gross revenue target ($) used by vessels in their return to port decision. If the value of this variable is zero or less, the vessel uses the expected net revenue method for this decision instead of the revenue target method (Section 4.3.7).
Vessel status	currentLocation	The grid cell that the vessel is in.
	currentActivity	A set of discrete states that describe the vessel's current activity (waiting in port, traveling, fishing, etc.). These states are defined in Section 2.3.3.
	departureDate returnDate	Dates on which the current (or next, when in port) trip started and is planned to end.
	setGearType	The gear type (TUNA or SWF) currently in use, or planned for the next set.
	totalFish[4]	The amount (lb) of fish currently stored.
	ageOfCatch	The time (days) since the first fish on board was caught.
	fishPrice[4]	The prices ($/lb) expected for the current catch.
	tripNumber	A unique (across vessels) identification number for the fishing trip the vessel is currently on, when the vessel is not in port.
	setHistory	A list of the fishing sets (including the information in Table 4) made so far in the fishing trip.
	travelTime	The time (whole days) needed to travel from port to its destination, when the vessel is traveling to or from a trip destination.
	fuelRemaining	The amount (lbs) of fuel currently in the vessel's tank.
	destinationLocation	For vessels traveling to start fishing, the cell they are traveling to.
	tripCost	The total cost (including fuel, wages, etc.) of the vessel's current fishing trip.

[1]There are actually separate variables for each gear type.
[2]Separate variables for each fish species caught, for each gear type.
[3]Separate variables for each turtle species.
[4]Separate variables for each catch species.

Table 4. Fishing set variables.

Variable	Description
vesselID	The vessel that made the fishing set.
tripNumber	The value of the vessel's variable tripNumber when the fishing set was made.
setGearType	Gear type of the fishing set (TUNA or SWF).
setDate	Date when the set was made. (Sets are assumed not to take more than one day to complete.)
longitude	The longitude (degrees) where the set was made.
latitude	The latitude (degrees) where the set was made.
cellID	The cell where the set was made.
fishCatch[1]	The catch of fish (lb).
turtleInteractions[2]	The number of turtle interactions occurring in the fishing set.
hooks	The number of hooks on the longline used for the set.
mainline	The length of main line (statute miles).

[1]Separate variables are used for each catch species.
[2]Separate variables are used for each turtle species.

2.2.4 Social networks

Social networks are "collectives" (Section 3.10) of vessels that share information on fishing locations and stock abundance. All vessels with the same value of their "network" variable belong to the same network and share information (a collective common memory) of past fishing success. The number and size distribution of social networks are controlled by the values of "network" given to vessels. At one extreme, each vessel can act independently (no social networks) if vessels all have unique values of "network". At the other extreme, if all vessels have the same value of "network" they all share information with each other. Between these extremes, there can be any range of network sizes, and the size distribution of networks can range from even (networks have similar numbers of vessels) to skewed (a few large networks and many small ones). In the current version, the ownership-based network is implemented.

A network has only one state variable, its "vesselMemory". This memory is a list of the grid cells that vessels in the network have "memory" of and hence can estimate fishing success and turtle interaction probabilities. Vessels consider only the cells in their network's vesselMemory when deciding where to fish (Section 4.3.2). The list represents social networks by assuming, in initializing and updating the list, that vessels within networks share information (and hence "memory") about cells they have fished in.

2.3 Process Overview and Scheduling

This section of the Overview summarizes the model's "actions", otherwise known as the events that occur during a simulation. Actions are described by (a) what real process the action represents; (b) which entities (or entity) execute the action and, in what order; and (c) what state variables are changed by it. Simple actions are described fully in this section, but the more complicated processes are described elsewhere in detail (Section 4.3)

The FMMHLF uses a kind of scheduling called "discrete event": instead of all model entities executing the same set of actions once per simulated day, some entities decide which action they will execute next and when. The following is sub-sections describes the potential actions and how entities (especially fishing vessels) decide which actions to execute, and when.

On each simulated day, the model executes the following groups of actions, in the order they are described below (HLF actions, then grid cell actions, then fishing vessel actions, etc.).

2.3.1 HLF actions

The HLF (Section 2.2.1) executes the follow actions.

1. Increment date: The date is increased by one day.
2. Update fish prices: Set the daily market price variable (priceToday) for each catch species. Daily prices are simply read in from an input file. (The Hawaii application currently uses the same daily prices for every year. These prices were estimated from historical data.)
3. Reset closures: If the current date is the first of January, then any fisheries closed due to effort or catch quotas, or turtle interactions, are re-opened. The HLF variable fisheryIsClosed is set to FALSE for all ocean regions and gear types.
4. Check date closures: The closures defined in variable dateClosures are checked. For each closure, if the current date is between the closure's starting and ending days then cells within the polygon are closed to fishing (cell variable fisheryIsClosedByDateClosure is set to TRUE) with the closure's gear type. "Within the polygon" means the cell's center is on or within the polygon's boundaries.
5. Check effort quota closures: Determine whether to close the fishery for each gear type because the limit on fishing effort (HLF state variable effortLimit) is exceeded by the sum of the effort expended by all vessels over the current calendar year. If so, the HLF variable fisheryIsClosed for the gear type and all ocean regions is set to TRUE.
6. Check catch quota closures: Determine whether to close the fishery for each gear type and region because the limit on catch is exceeded. For each catch species, the sum of catches by all vessels, in the current calendar year, is calculated for each ocean region. If that sum exceeds the quota in an ocean region (HLF state variable catchQuota), the HLF variable fisheryIsClosed for the region and for the gear type that targets the catch species is set to TRUE. (Hence, by-catch contributes to a catch quota being exceeded, but when a catch quota is exceeded the catch species could still be caught as by-catch of gear types that do not target it.)
7. Check turtle interaction closures: Determine whether to close the SWF fishery in all regions due to turtle interactions. The value of fisheryIsClosed for the SWF gear type is set to TRUE if the limit on turtle interactions (HLF state variable turtleCap) for any turtle species is exceeded by the sum of turtle interactions for that species, over all vessels within the current calendar year. These closures apply to all ocean regions.

2.3.2 Grid cell actions

All cells execute the same assumed actions daily. The order in which they execute these actions is arbitrary because it has no effect.

8. Update CPUE: If the current date is the first day of a month, then the state variables CPUEMonthly are also updated as described in Section 4.3.1. On every day, the state variables CPUEDaily for each catch species are updated, using the method described in Section 4.3.1.

9. Update IPUE: If it is the first day of a month, then the IPUEMonthly variables are updated as described in Section 4.3.1. (IPUEDaily is assumed equal to IPUEMonthly.)

2.3.3 Fishing vessel actions

Vessel agents carry out fishing decisions and activities. These actions can be thought of as an event tree, where what a vessel does depends on what happens to it within a simulated day. Which actions a vessel executes first depends on its activity state (state variable currentActivity) at the beginning of a time step. Figure 2 provides an overview of vessel actions and activity state changes.

The following subsections define the schedule of vessels in each activity state. It lists the actions for in each state, in the order they are executed. If the description of an action does not say explicitly which action is executed next, or that no further actions are executed on the current day, then the following action is executed next.

The order in which vessels execute these actions is randomized each daily time step. When a vessel starts executing its daily actions, the first action executed is the first one for its activity state. For example, if a vessel was in state "travelingToPort" at the end of the previous day, its first action on the current day is "Determine arrival" (Section 2.3.3.5).

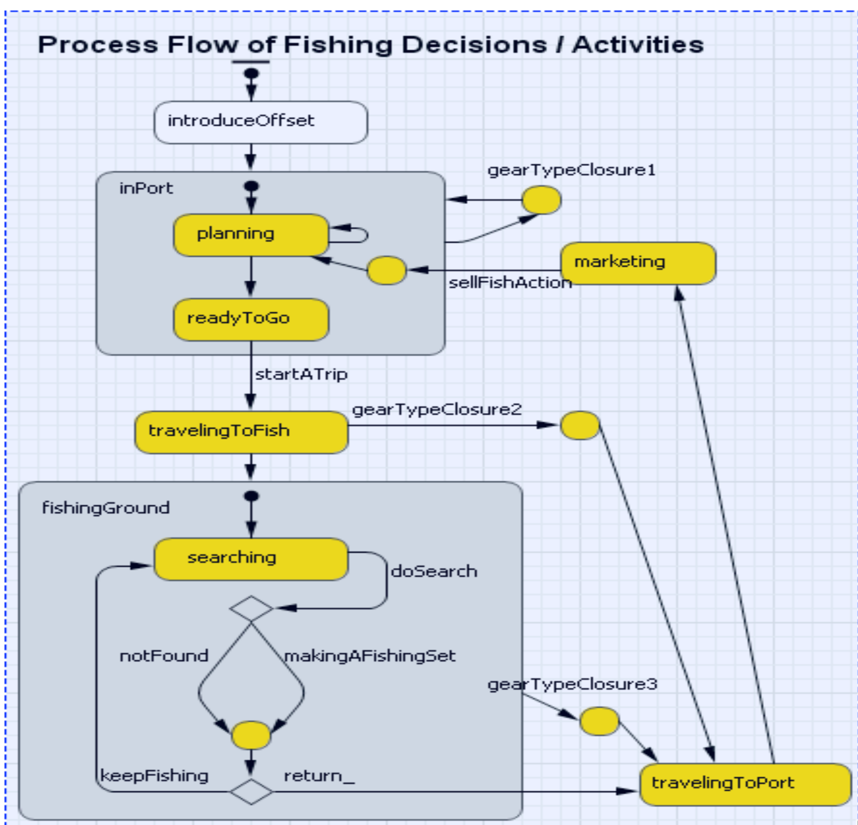


Figure 2. Overview of fishing vessel states and decisions.

2.3.3.1 Vessels with current activity of "inPort"

10. Respond to closure: This action is executed only if the vessel has already decided to make a fishing trip (below) but has not yet left port, and fishing with its gear type has been closed in the ocean region of its destination cell (via the corresponding HLF variable fisheryIsClosed). If so, the vessel executes its fishing trip decision (Section 4.3.2) again, to plan a new trip that considers the closure.
11. If the vessel has already decided to make a fishing trip (see following action) but has not yet left port it executes no further actions on the current day. If the vessel has not yet made a fishing trip decision since arriving in port, or if the number of days since last making a fishing trip decision exceeds the value of parameter planningHorizon, then the following action is executed.
12. Make fishing trip decision: The vessel decides, using methods detailed in Section 4.3.2, whether to stay in port or start a new trip. If a trip is to be started, this action also determines the gear type and destination cell (vessel variables setGearType and destinationLocation), and the date on which it departs. No further actions are executed for the day.
13. Start a trip: This action is only executed if the vessel's previous fishing trip decision is to start a new trip and the number of days between the current day and the day when the decision was made equals or exceeds restingTime. The vessel's tripNumber variable receives a new unique value. The variable travelTime is set to the distance to the destination cell divided by the vessel's speed. The value of fuelRemaining is set to fuelCapacity, assuming vessels always leave completely fueled. The vessel's value of currentActivity is changed from "inPort" to "travelingToFish". The travelingToFish actions (Section 2.3.3.2) are then executed.

2.3.3.2 Vessels with current activity of "travelingToFish"

14. Respond to closure: The vessel determines whether fishing with its gear type has been closed in the ocean region of its destination cell (via the corresponding HLF variable fisheryIsClosed). If so, it executes the response to closure as described in Section 4.3.3.
15. Determine arrival: The vessel determines whether it has arrived at its destination cell, by comparing the time it has been underway (the difference, in whole days, between the current date and the date on which it set its currentActivity to "travelingToFish"; so the lowest value of time underway is 1 day) to its value of travelTime. If time underway equals or exceeds travelTime, the vessel is assumed to have arrived: it changes its activity to "searching", and executes the following action. If time underway is less than travelTime, the vessel executes nothing else for the day.
16. Update trip variables. This action is executed only if the vessel arrives at its destination in the previous action. The variable tripCost is reset and updated by setting it to travelTime × travelCost. The value of fuelRemaining is decreased by travelTime × fuelPerDay. The vessel's currentActivity is changed to "searching" and it executes no further actions for the day.

2.3.3.3 Vessels with current activity "searching"

17. Respond to closure: The vessel determines whether fishing with its gear type has been closed in the ocean region of its current cell (via the corresponding HLF variable

fisheryIsClosed and the cell variable fisheryIsClosedByDateClosure). If so, it executes the response to closure action described in Section 4.3.3.

18. Make fishing set decision: The vessel executes the fishing set decision submodel, in which it decides whether to fish and which cell to fish in, using a decision method based on net returns (Section 4.3.4). If the vessel decides to fish on the current day, it changes its activity to “makingAFishingSet” and executes the actions for that activity. If not, it keeps its activity equal to “searching” and executes the following actions.
19. Update trip variables: The variable tripCost is increased by the daily cost of traveling (vessel variable travelCost). The fuel remaining is updated by subtracting daily fuel use (variable fuelPerDay), and the ageOfCatch variable is increased by a day *if* any fish have been caught on the current trip.
20. Decide whether to return to port: At the end of each day of searching, the vessel is assumed to determine whether it will search again the next day or return to port. This decision is detailed in Section 4.3.7. If the vessel decides not to return, its current activity remains “searching” and it executes no further actions for the day. If it does decide to return to port, the vessel executes the following action.
21. Initialize return trip: This action is only executed if the vessel decides to return to port. The variable travelTime is set to the distance to port from the current cell divided by the vessel’s speed. The vessel’s value of currentActivity is changed from “makingAFishingSet” to “travelingToPort”. No further actions are executed for the day.

2.3.3.4 Vessels with current activity “makingAFishingSet”

22. Respond to closure: The vessel determines whether fishing with its gear type has been closed in the ocean region of its current cell (via the corresponding HLF variable fisheryIsClosed and the cell variable fisheryIsClosedByDateClosure). If so, it executes the response to closure action described in Section 4.3.3.
23. Update trip variables: The variable tripCost is increased by the daily operating cost (variable operatingCost). The fuel remaining is updated by subtracting daily fuel use (vessel variable fuelPerDay), and the ageOfCatch variable is increased by a day if any fish have been caught on the current trip.
24. Conduct fishing set: The vessel conducts a fishing set, using methods detailed in Section 4.3.5.
25. Determine turtle interaction: The vessel determines, using methods described in Section 4.3.6, whether or not it interacts with a turtle. If so, it reports the interaction by incrementing the value of the HLF variable turtleInteractionsSoFar for the species of the turtle the vessel interacts with.
26. Decide whether to return: At the end of each day of fishing, the vessel is assumed to determine whether it will fish again the next day or return to port. This decision (detailed in Section 4.3.7) is modeled as a financial decision that considers the vessel’s current state (fish storage capacity; fuel) and fishing success so far on the trip. If the vessel decides not to return, it changes its current activity to “searching” and then executes no further actions on the current day. (Hence, only one fishing set can be made per day.)
27. Initialize return trip: This action is only executed if the vessel decides to return to port. The variable travelTime is set to the distance to port from the current cell divided by the vessel’s speed. The vessel’s value of currentActivity is changed from “makingAFishingSet” to “travelingToPort”. No further actions are executed for the day.

2.3.3.5 Vessels with current activity of "travelingToPort"

28. Determine arrival: The vessel determines whether it has arrived at its port, by comparing the time it has been underway (the difference, in whole days, between the current date and the date on which it set its currentActivity to "travelingToPort"; so the lowest value of time underway is 1 day) to its value of travelTime. If time underway equals or exceeds travelTime, the vessel is assumed to reach port. Otherwise, the vessel executes no further actions for the day.
29. Record trip variables. This action is executed only if the vessel arrives at port in the previous action. The variable tripCost is increased by travelTime × travelCost. The value of fuelRemaining is decreased by travelTime × fuelPerDay. The ageOfCatch variable is increased by travelTime. The vessel's currentActivity is changed to "marketing" and no further actions are executed for the day (vessels are assumed to market their fish the morning after arrival in port).

2.3.3.6 Vessels with current activity of "marketing"

30. Sell fish: The vessel sells its catch, using methods detailed at Section 4.3.7.2. It then changes its activity to "inPort" and executes no other actions on the current day.

2.3.4 Output

Output from FMMHLF includes two primary output files, and several optional output files. (The model's software also provides graphical output updated as it executes, displaying which vessels are where and what activity they are conducting.) Many simulation models write summary output only at the end of each time step. Some output from FMMHLF, however, is reported for each individual vessel instead of being summarized, and is written throughout the schedule as the actions affecting key results are executed. The two primary output files report are as follows.

- Fishing sets. Each time a fishing set is completed, the information in Table 4 is written as a line of results in the fishing set output file.
- Fishery status. At the end of each simulated day, a line is written to the fishery status output file reporting key HLF variables (Table 1) such as the date, number of trips and sets made so far, amount and value of fish caught, and number of turtle interactions. The number of trips is reported separately for each ocean region; the region assigned to each trip is that of the vessel's destination cell at the start of the trip. The number of trips output is the number of trips *started*, so includes trips currently in progress. The total weight of fish caught is updated as fish are caught, and the total value of the catch is updated when fish are sold; hence, fish caught but not yet marketed (in storage aboard vessels) are reflected in the catch output but not the catch value output.

The optional output files were designed for testing the model software and detailed analysis of model results. They report inputs to, and intermediate and final results of all submodels (Section 4.3). Optional output files also report detailed information such as the location and state of each vessel over time, and the state of each grid cell over time.

3 DESIGN CONCEPTS

3.1 Basic Principles

The basic type of theory used in FMMHLF is financial decision-making: the important behaviors are modeled by assuming individual fishing vessel operators use typical financial methods such as net revenue analysis. What makes the model novel is that each vessel makes different decisions that depend on its characteristics and state, and that environmental regulations (potential closures for sea turtle protection) are treated as part of the financial decision.

3.2 Emergence

The main results of FMMHLF are:

- Distributions of fishing effort and catch over space (e.g., ocean regions or fishery management areas), time (e.g., months or seasons), and between the swordfish and tuna fisheries; and
- Changes in the above distributions in response to turtle protection regulations.

The model is designed so that those results emerge from:

- Historical data on catch success and turtle interactions over space;
- Market prices for fish;
- Fishing vessel characteristics, using a distribution of characteristics designed to resemble a real longline fleet;
- Regulations for sea turtle regulation and the rate of interactions between vessels and turtles;
- How vessel operators adapt to the above as they decide: when, where, and with what gear (swordfish vs. tuna) to fish; and how many sets to fish each trip.

3.3 Adaptation

The only entities in the model with adaptive behavior are the fishing vessels. They make three adaptive decisions. (1) Fishing trip decision: when in port, deciding when, where, and with which gear to next fish. (2) Fishing set decision: when on a fishing trip, deciding where (which cell) to make the next fishing set. (3) Return to port decision: when on a fishing trip, deciding whether to fish another day or return to port. These decisions are all modeled as explicit objective seeking: the vessels choose the alternative providing highest value of an objective function that represents the vessel's utility (net income). Alternatively, the vessels could seek to reach a certain target revenue level for a fishing trip. The two assumptions could be switched as a part of input parameters. The current version uses the assumption of profit-maximization as the default.

3.4 Objectives

For the fishing trip decision, the objective measure represents expected daily mean net return during the trip. The objective considers expected income from fish harvest and trip costs. Unlike the objectives for other decisions, this measure includes the potential effect of fishery closures due to turtle interactions on net return. The objective measure considers only the current trip; it does not consider how the decision affects the number and timing of future trips.

The fishing set decision objective is to maximize expected net revenue for the current day, given the alternatives of fishing in one of the adjacent cells or not fishing. The objective considers only fish catch value and operating costs.

The objective measure for the return to port decision is to maximize expected net revenue for the following day, for the alternatives of fishing another day or returning to port. The measure considers revenue from additional fish catch, operating costs, and the potential cost of previously caught fish becoming too old to sell.

3.5 Learning

Learning is the process of changing the adaptive behaviors (Section 3.3) through experience. Captains and owners of vessels get to know the CPUE of a cell by making fishing sets in the cell which will affect their fishing decisions subsequently. When simulation first starts, each vessel is initialized with a unique set of cells with per-defined CPUE (values are estimated from the HLF data) where it had made fishing sets previously, i.e., memory. It then starts to update (reveal) the CPUE of the cells as fishing activities are simulated (Section 4.3). Leaning is reflected in two aspects: 1) CPUE of a cell that previously is not in a vessel's memory could be added and become a potential fishing cell for the vessel once it makes a fishing set in it; 2) the pre-defined CPUE of a cell will be updated with the CPUE estimated from the catches of the fishing sets that are made by the vessels in the cell.

3.6 Prediction

In their adaptive decisions, vessels rely on prediction of future fish catch, market prices, and whether fisheries will be closed due to the quota on turtle interactions.

Predictions of catch for the fishing trip decision assume vessels know the actual catch rate they will experience, but only the monthly mean catch rate and not daily variation around that mean. For the fishing set and return to port decisions, vessels are assumed to know the actual catch they will experience.

Predictions of fish market prices for the fishing trip decision assume vessels know the average price over a range of dates around the time they would market their catch. This assumption reflects some ability of vessels to anticipate seasonal market variation but not day-to-day variation. The fishing set and return to port decisions predict fish prices by assuming they are equal to the day's current price.

In the fishing trip decision, vessels predict the probability of a closure due to turtle interactions. This probability is modeled as a Poisson process, predicting the probability as a function of the rate of turtle interactions so far in the year, the date a trip would start, and the trip's expected duration.

3.7 Sensing

Vessel sensing of fish catch and potential turtle interactions are modeled as a memory-based process. Vessels are assumed to know the fish catch rates (CPUE) and turtle interaction rates (IPUE) of cells they might fish in, but only in cells they have information on because those cells have been fished in by vessels in the same social network.

Vessels are also assumed to know historical market prices for fish and the possible average market in the near future; this assumption is realistic because average market prices are readily available to real fishing vessels.

3.8 Interaction

There are two kinds of interaction among vessels. First, vessels in the same social network (with the same value of vessel variable "network") contribute to and share a common memory of fishing conditions in cells (Section 2.2.4). The second interaction among vessels is a subtler one: in the fishing cell selection decision, vessels avoid cells already containing a number of other vessels specified by the vessel parameter maxVesselsPerCell. Because depletion of fish is not represented, there is no interaction via competition for catch.

3.9 Stochasticity

Stochasticity is incorporated in the simulation here to model the randomness in environmental conditions, and decision outcomes. Stochasticity is used in the following ways:

- During model initialization, the date on which each vessel start executing their actions is chosen randomly. This is to keep vessels from behaving synchronously.
- The order in which vessels execute their actions is randomized daily to avoid artifacts of execution order.
- The daily CPUE in each cell includes stochastic variation, to represent realistic variability in catch.
- When vessel decisions produce ambiguous results (e.g., more than one alternative has the best net revenue; no cells are profitable for fishing), a choice among equally good alternatives is made randomly.
- Turtle interactions are modeled as random processes.

3.10 Collectives

Social networks (Section 2.2.4) are collectives of vessels: groups of vessels that are affected by their member vessels (which contribute to the social network's common memory of fishing cells) and affect their members (which each use the common memory in fishing decisions). These collectives are represented as explicit entities with one variable (the memory) and no behaviors.

3.11 Observation

Observations of the model are described at Section 2.3.4.

4 DETAILS

The "Details" section describes how the model is initialization with the HLF parameters and how sub-models constitute the FMMHLF model. It provides the underlying mathematical equations/formulas that derives the codes constructed the simulation model.

4.1 Initialization

This section describes how model entities are given initial values of their state variables at the start of a simulation. The methods used to develop initialization data for the Hawaii fleet are documented in Appendix A.

4.1.1 HLF

Table 5 describes how HLF variables (defined in Table 1) are initialized.

Table 5. Initialization of HLF variables.

Variable	Initial value
date	1 January of an unspecified year.
numberOfVessels	0
numberOfTrips	0
numberOfSets	0
catchSoFar	0
catchValue	0
turtleCap	(Input as model parameters)
turtleInteractionsSoFar	0
catchQuota	(Input as model parameters)
effortLimit	(Input as model parameters)
fisheryIsClosed	False
priceToday	Variables are set by first day's HLF actions; Section 2.3.1.

4.1.2 Cells

The static state variables of cells (Table 2) are read in as input. The dynamic variables (e.g., for fishing success and turtle interaction rates) are initialized by the grid cell actions (Section 2.3.2) of the first simulated day.

4.1.3 Vessels

The static state variables of vessels (Table 3) are initialized to create a virtual fleet with characteristics representative of the actual longline fishery. Values of each static variable for each vessel are read from an input and the number of vessels is determined from the input file. For the Hawaii application, these initialization data were developed from log book data of actual vessels, so that the model's distribution of vessel characteristics (especially size and gear types) represent the real longline fleet, while not allowing simulated vessels to be associated with particular real vessels. The following state variables are initialized to define the characteristics of each vessel: size, vesselType, fuelCapacity, fishCapacity, fuelPerDay, hooks, speed, fishHoldingCapacity, maxDaysInSea, operatingCost, travelCost, CPUE-VesselFactor, priceAdjFactor, IPUE-VessselFactor. The dynamic vessel state variables are initialized during execution of vessel actions.

To de-synchronize vessel actions at the start of a simulation, vessels do not start executing their schedule (Section 2.3.3) until a randomly chosen day. This day is drawn (separately for each vessel) from a uniform integer distribution ranging from 1 to the value of a HLF parameter vesselInitializationDateSpread, which has a standard value of 60 as indicated from the HLF data. Hence, vessels are added to the schedule over simulation days 1 to 60. When a vessel starts executing, its activity state is set to "inPort" (Section 2.3.3.1) with its fishing trip decision unmade.

4.1.4 Social networks

The vesselMemory variable (a list of cells that vessels in the network know about for making fishing trip decisions) is initialized from an input file. The file identifies all the grid cells that each vessel has memory of at the start of a simulation. Each such cell is added to the vesselMemory list of the social network that the vessel belongs to (unless it is already on the list).

4.2 Input data

In the ODD protocol, "input data" refers specifically to data read into a model to represent variation over time in state variables, usually representing the agents' environment. FMMHLF currently uses only one kind of time-series input data.

4.2.1 Fish prices

The HLF state variables priceToday for each fish catch species are updated daily from input.

4.3 Submodels

4.3.1 Cell catch and turtle interaction rate updates

The value of each cell's variables CPUEDaily and IPUEDaily are updated. While this action is executed each day, some calculations are executed only on the first day of each month.

Separate values of CPUEDaily are calculated for each combination of gear type and catch species. (Including all combinations allows for "by-catch", by representing the catch of each species even when it is not targeted by the gear type in use.)

The value of CPUEDaily (lbs fish per 1000 hooks) is drawn from a random normal distribution, the mean of which (CPUEMean) is updated on the first day of each month. CPUEMean is calculated as:

$$CPUEMean = constantCPUE + monthlyCPUE + yearlyCPUE + gridCellCPUE$$

where constantCPUE is a HLF parameter, monthlyCPUE is a HLF parameter with values for each calendar month, yearlyCPUE is a HLF parameter with values for each year type (HLF parameter baseYear), and gridCellCPUE is a parameter provided for each cell. These parameters are intended to be evaluated from historical fishing data. (Each of these parameters is actually a set of parameters, with one value for each combination of gear type and catch species.)

If CPUEMean is positive, the cell's CPUEDaily is updated each day by drawing a sample from a random normal distribution with mean CPUEMean and standard deviation of 100 lbs fish per 1000 hooks; if the sample is negative then CPUEDaily is set to zero. However, if CPUEMean is negative then CPUEDaily is simply set to zero; this rule allows (for example) cells with no historical catch data to be represented as having no potential catch by setting gridCellCPUE to a large negative value such as -9999.

IPUEMonthly (probability of a turtle interaction, per 1000 hooks; however, the probability experienced by each vessel is adjusted for vessel characteristics; Section 4.3.6) is also modeled as a function of cell and date. Separate values of IPUEMonthly are calculated for each combination of gear type and turtle species. (The Hawaii application uses parameter values assuming IPUEMonthly is zero for tuna gear.) The value of IPUEMonthly is calculated as:

$$IPUEMonthly = monthlyIPUE + gridCellIPUE$$

where constantIPUE is a HLF parameter, monthlyIPUE is a HLF parameter with values for each calendar month and cell, and gridCellIPUE is a parameter provided for each cell. These parameters are intended to be evaluated from historical fishing data. (There are actually parameter values for each combination of gear type and turtle species.)

In this version of FMMLHF, daily variation in IPUE is not represented, so IPUEDaily is set to IPUEMonthly.

4.3.2 Fishing trip decision

The fishing trip decision submodel is executed by vessels when in port (vessel activity is "inPort"). It is executed on the day the vessel first enters the simulation schedule at the start of a model run (Section 4.1.3) and on the day after the vessel has marketed fish from its previous trip; and can also be triggered by several other events in the vessel schedule. The submodel sets several vessel variables (Section 2.2.3) that represent trip planning: departureDate and returnDate, setGearType, destinationLocation, and travelTime. Within the submodel, the variable "tripDelay" is the number of days the boat stays in port, after making this decision, before leaving for the next fishing trip. Vessels delay their departure if they expect increases in catch rates or fish prices sufficient to offset the loss of fishing days; tripDelay also includes a "resting period" to re-equip the vessel between trips.

The objective of the trip decision is to maximize expected net returns, as a daily mean return ($ profit per day between the current day and end of trip). The return is weighted by the probability of the fishery not being closed due to turtle interactions, so it is referred to as "Interaction-weighted Net Daily Return" (INDR). Simplifying assumptions include: (1) For planning, the trip duration is always assumed equal to the vessel's average trip duration (vessel variable typicalTripLength); (2) Vessels plan fishing trips one at a time, considering only their expected return from that trip; and (3) Vessels ignore any effect their decision could have on turtle interactions (they do not base their choice of fishing gear or location on how likely they are to interact with turtles).

Vessels execute the submodel by calculating INDR for all alternatives available to them, and selecting the alternative with highest INDR. The alternatives available to a vessel are all combinations of:

- departureDate, which can range from the end of the resting period (tripDelay = restingTime, defined below) to the current date plus the parameter planningHorizon (tripDelay = planningHorizon).
- gear type; the gear type options for a vessel are determined by its variable vesselType. However, a gear type is not considered if its use is currently closed in all cells in the model (due to the turtle interaction quota being exceeded).
- cell: a vessel considers all cells in its memory (the variable vesselMemory of the vessel's social network), except those expected to be closed to fishing with the gear type. Cells are expected to be closed if either (1) they are currently closed due to the HLF's quotas for effort, catch, or turtle interactions; or (2) they are scheduled to be closed due to date closures

(Section 2.3.2) on any days between the first and last days the vessel expects to fish (explained below).

The minimum number of days between the end of one trip (the day fish are marketed) and the start of another trip is restingTime (d). The value of restingTime represents the time a vessel takes to re-stock with supplies, etc., before leaving again. This value is assumed to differ with the gear type of the upcoming trip. For each trip decision of each vessel, the value of restingTime is an integer drawn randomly from a negative binomial distribution with *n* parameter equal to the HLF parameter *restingTimeN* and *p* parameter equal to *restingTimeP*. Separate parameters are used for planning each gear type. If this randomly drawn integer is less than 1 it is set to 1. If the value of restingTime exceeds planningHorizon then tripDelay is set equal to planningHorizon. INDR is calculated as:

$$INDR = \left(\frac{tripRevenue - tripCost}{tripDelay + tripDuration}\right)(expectedPNotClosed).$$

Several terms in this equation depend on how many days the trip under consideration lasts and how many of those days would be spent fishing (the variable daysFishing) vs. traveling. For the planning decision, the total trip duration (tripDuration) is assumed equal to the vessel variable typicalTripLength. The number of days spent fishing (daysFishing) is assumed equal to typicalTripLength minus the number of days spent traveling to and from the cell under consideration, minus also one day assumed for searching (Section 2.3.3.3). The number of days traveling to and from a cell (daysTraveling) is calculated as the distance from port to cell (calculated as explained in Section 2.2.2) divided by the vessel's speed, multiplied by two (for the return trip) and rounded to an integer. If the resulting value of daysFishing is zero or negative, then INDR is set to zero and the rest of its calculation is skipped. The date in the middle of fishing is used in this submodel; it is calculated as the departure date plus the time to travel to the cell plus one (for searching) plus one-half of daysFishing, all rounded to an integer date.

TripRevenue is the expected gross income from fishing on the trip. This expected income is based on all the catch species, including bycatch, not just the species targeted by the gear type under consideration. The incomes from the catch species are calculated separately and then summed. TripRevenue is calculated from variables of the vessel and cell:

$$tripRevenue = estimatedCPUE \times hooks \times daysFishing \times expectedPrice.$$

EstimatedCPUE is the vessel's estimate of average catch per 1000 hooks during the trip. It is simply set to the value the cell's variable CPUEMonthly will have on the day midway between departureDate and returnDate.

ExpectedPrice is the vessel's estimate of its catch's market price upon its return to port. Vessels are simply assumed to know the market price they would obtain, averaged over a range of dates around their expected date of returning to port. The range of dates is defined by vessel variable priceExpLength (which has a value of 3 for the Hawaii application). ExpectedPrice is therefore the mean of the prices that the vessel would get (using the price calculation methods in the fish sale submodel; Section 4.3.7.2) for the catch species corresponding to the gear type, over dates starting at (returnDate – priceExpLength) and ending at (returnDate + priceExpLength).

TripCost is the vessel's estimated cost of making the fishing trip, estimated from vessel cost variables and expected trip length:

$$tripCost = (daysTraveling \times travelCost) + (daysFishing \times operatingCost)$$

where operatingCost is for the gear type being considered.

The final term in the equation for INDR is expectedPNotClosed, the expected probability that fishing anywhere with the gear type will not be closed due to turtle interactions. This probability depends on the HLF variables turtleCap and turtleInteractionsSoFar (sections 2.2.1, 2.3.1). If the gear type under consideration is not subject to turtle interaction closures (TUNA, in the Hawaii application), then expectedPNotClosed is simply 1.0. Otherwise, the probability is modeled as a Poisson process for which events are turtle interactions, the Poisson distribution models number of interactions per day, and the distribution's rate parameter (interactionRate, mean number of interactions per day for the calendar year to date) is calculated by dividing turtleInteractionsSoFar by the Julian date of the day on which the fishing trip decision is made. ExpectedPNotClosed is modeled as the probability that the number of turtle interactions between the current date and the middle day of the fishing trip is less than the number of interactions remaining in the quota (interactionsRemaining = turtleCap – turtleInteractionsSoFar):

$$expectedPNotClosed = \mathrm{Poisson}(interactionsRemaining, daysToMidTrip \times interactionRate)$$

where daysToMidTrip is the number of days from (and including) the current day until (and including) the expected middle day of fishing on the trip (defined above). If interactionRate is zero because turtleInteractionsSoFar is zero, then expectedPNotClosed is set to 1.0.

When more than one species of turtle is protected by separate interaction quotas, expectedPNotClosed is calculated separately for each turtle species; these species-specific values are then multiplied together to produce the probability that the interaction quotas of no turtle species will be exceeded.

After calculating INDR for all alternative combinations of departure date, gear type, and cell, the vessel identifies the combination providing highest value of INDR. (If two or more combinations offer the same value of INDR, one is selected randomly.) If none of these alternative trips provide a positive value of INDR, then the vessel decides to stay in port for the duration of its planning horizon: the vessel variables defining a trip (for destination cell, gear type, etc.) are not updated and the vessel's value of currentActivity status remains "inPort". Otherwise, the vessel's values of departureDate and returnDate are set to the departure date and expected return date of the best alternative, destinationLocation is set to the best alternative's cell, and setGearType is set to the gear type of the best alternative. The vessel variable travelTime is set to the time for traveling from port to the cell, calculated as the distance (Section 2.2.2) divided by the vessel's speed.

As an example, the following figures illustrate key dynamics of the fishing decision submodel. They were developed assuming SWF gear, a typical trip length of 20 d (an average value for vessels with SWF gear), two days travel time to the cell, and travel and operating costs of $490 and $450/d (also typical for Hawaii's SWF vessels). Except as varied in the figures, CPUE is 2000 lbs/1000 hooks (typical for cells with SWF catches in the Hawaii historical records), and swordfish price is $2.36/lb (the average for one historical year). The figures assumed one turtle species with a quota of 21 interactions per year. .

The value of INDR varies approximately equally with both fish price and CPUE, when turtle interactions are neglected (Figure 3). For the assumed vessel and trip characteristics, SWF trips are profitable over wide ranges of price and CPUE.

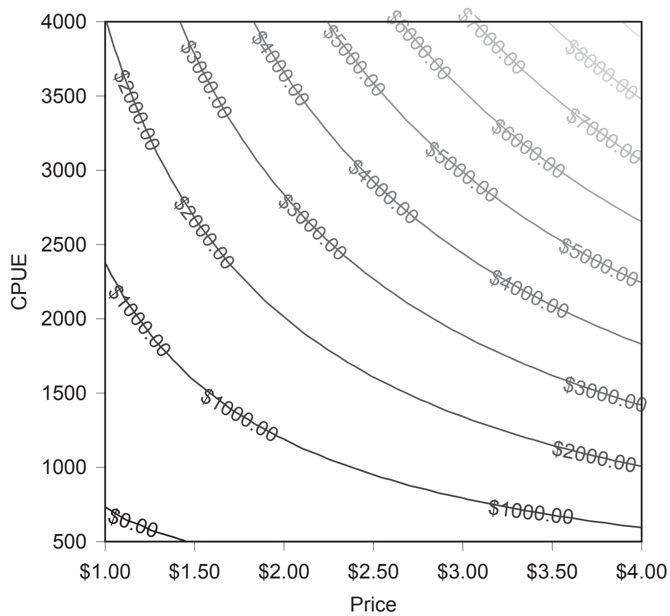


Figure 3. Contours of Interaction-weighted Net Daily Return (INDR) vs. vessel's CPUE and price for SWF, assuming no probability of fishery closure due to turtle interactions.

Turtle interactions affect trip decisions only when the rate of turtle interactions is high and the quota on the number of interactions is almost met. When the number of turtle interactions as of a June 1 trip departure date was varied up to 20, assuming the fishery is closed when 21 turtle interactions occur, the expected profitability of the trip was affected only when >17 interactions had occurred (Figure 4). Trips remained profitable even at 20 interactions. When 20 interactions have occurred by June 1, the daily interaction rate is 0.13 and the probability of one more interaction over the 12-day period assumed for this example is slightly less than 50%.

When interactions are concentrated earlier in the year, they have stronger effects on trip decisions (Figure 5). This relationship is because high numbers of interactions early in a year result from a high daily rate of interactions, which produces a higher probability of more interactions during a fishing trip. However, when the quota is almost met even by early summer, fishing trips are still expected to be profitable.

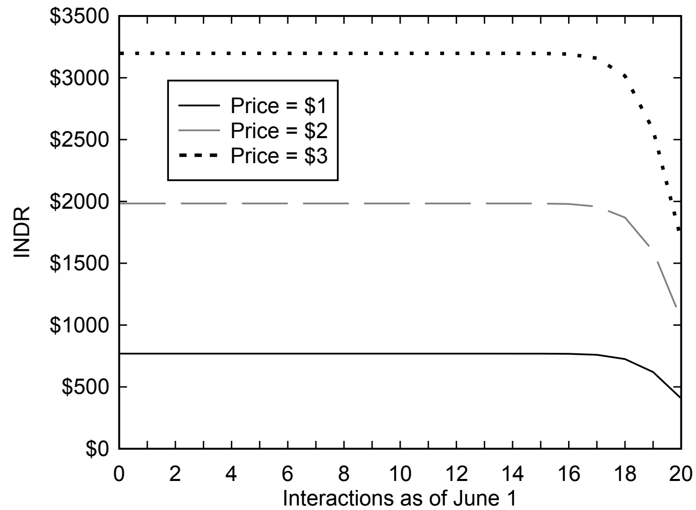


Figure 4. Interaction-weighted Net Daily Return (INDR) vs. number of turtle interactions, out of a quota of 21 interactions, at June 1. Three SWF prices are used.

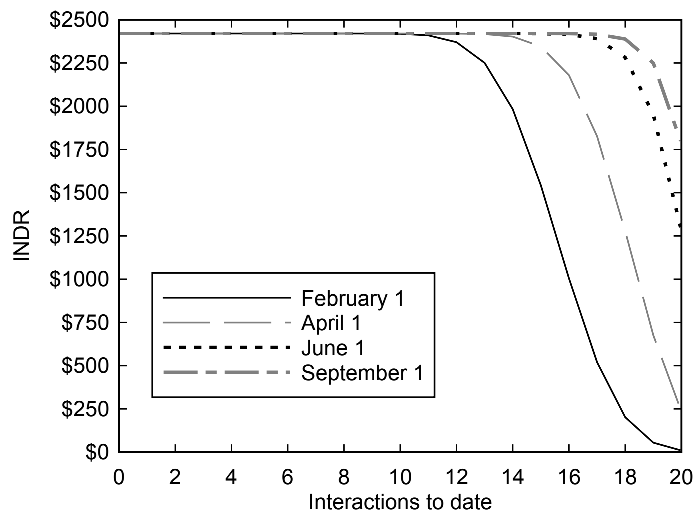


Figure 5. Interaction-weighted Net Daily Return (INDR) vs. number of turtle interactions, out of a quota of 21, for four trip departure dates.

4.3.3 Respond to closure

This submodel is executed by vessels when they are traveling to or conducting a fishing trip and the cell chosen for fishing becomes closed. (Vessels still in port respond to closures using a method described in Section 2.3.3.1.) When this happens, the submodel assumes vessels simply return to port to market any fish caught on the trip and plan a new trip. This assumption is made (instead of assuming that vessels search for alternative cells where fishing is still open and profitable) because the regulatory policies we attempt to examine in this simulation model (sea turtle cap, swordfish fishing set quota and big eye tuna catch quota) requires the fishermen to do so. The fishing trip decision submodel (Section 4.3.2) prevents vessels from planning trips to cells that will be in date closures during the trip. The more likely kinds of closures (due to fish catch or effort quotas, or turtle interactions) affect whole ocean regions or the entire fishery, making it unlikely that alternative cells could be found.

When the respond to closure submodel is executed, the vessel changes its activity status to "travelingToPort" and then executes the vessel actions for that status (Section 2.3.3.5). That means the vessel begins traveling to port, with traveling time equal to the current value of the travelTime variable. (TravelTime is for the distance between port and the planned fishing cell, which could be a day or two in error for vessels turning around partway to their fishing destination, but this error is neglected.)

4.3.4 Fishing cell selection

This submodel represents a vessel's decision of whether to fish on the current day, and which cell to move to either fish or wait. The decision is made to maximize expected net revenue from fishing. The alternatives assumed available to a vessel are to either (a) fish in the most profitable of the nine neighborhood cells (its current cell and eight surrounding cells) or (b) move to a random neighborhood cell if none are profitable and wait a day. (There will be fewer than eight surrounding cells if the vessel is on an edge of the simulated space.)

The vessel first determines which cells are potential fishing cells. Cells are potential fishing cells if (a) they are the vessel's current cell or adjacent to it, (b) they are not closed to fishing (cell variable fisheryIsClosedByDateClosure is not true; other types of closures have been considered in the vessel's "respond to closure" action), and (c) the number of other vessels that have already chosen to fish in the cell is no more than the vessel parameter maxVesselsPerCell, which has a standard value of 8. (The number of other vessels in a cell includes all vessels, no matter their current activity or whether they decided to fish in the cell on the current day. Hence, the order in which vessels execute this action will affect results.)

The second step is to calculate expected net revenue from one fishing set in each potential fishing cell. Expected gross revenue is the sum, over all catch species, of expected catch (lbs) times expected price ($/lb). Expected catch is calculated using the method to calculate catch in the fish catch submodel (Section 4.3.5). Expected price is calculated as in the fish sale submodel (Section 4.3.8) using the current day's value of priceToday.

Expected net revenue is then calculated as expected gross revenue minus the cost of one fishing set (vessel variable operatingCost, for the current gear type). The cost of waiting without fishing for a day (the vessel variable travelCost) is treated as an avoided cost in net revenue. Hence, expected net revenue is equal to expected gross revenue plus travelCost minus operatingCost. Fishing in a cell is considered profitable if expected net revenue is positive. (To be worthwhile, net revenue from a day of fishing need not be positive but must exceed the negative revenue of waiting without fishing.)

Finally, the vessel selects as its fishing cell the potential fishing cell offering highest expected net revenue. The vessel moves there and changes its activity to "makingAFishingSet". If more than one cell has the same, highest, net revenue then one of them is chosen randomly. However, if none of the cells offer positive net revenue, the vessel instead selects one of the potential fishing cells at random—excluding its current cell, if there are any others—and moves there; in this case the vessel does not change its activity from "searching".

4.3.5 Fish catch

This submodel calculates the catch (lbs of fish of each catch species, per set) from a fishing set. The catch is calculated separately for each species that the vessel's current gear type can catch (see Section 2.2.1). The catch of a species is a function of the cell's variable CPUEDaily and the vessel's CPUEAdjFactor variable for the vessel's current gear type and catch species, and of the vessel's variable hooks for its gear type. The catch is calculated simply as:

$$catch = (CPUEDaily + CPUEAdjFactor) \times hooks.$$

For each catch species, the catch is added to the vessel's variable totalFish (for the current gear type and catch species) and to the HLF's variable fishLanded for the catch species. The HLF's variable numberOfSets for the vessel's gear type is incremented by 1. The cell is also added to the vessel's memory of cells (the vesselMemory list of the vessel's social network) if not already on it.

As part of this submodel, data on the fishing set are recorded. A new fishing set record is created in the vessel's list of sets, and the variables in Table 4 are recorded in it. Quotas are not checked when the fish catch is simulated; they are checked once per day as part of HLF actions (Section 2.3.1).

4.3.6 Turtle interaction

This submodel is executed once per fishing set to determine whether the set produced an interaction with a turtle of any turtle species. (The possibility of one vessel interacting with more than one turtle of the same species, on a single fishing set, is neglected.) For each turtle species, whether an interaction occurs is modeled as a random Bernoulli trial with probability true (probInteraction) depending on variables of the cell and vessel:

$$probInteraction = IPUEDaily \times IPUEAdjFactor \times hooks$$

This value is calculated using the cell's value of IPUEDaily corresponding to the turtle species and the vessel's gear type, the vessel's value of IPUEAdjFactor for the turtle species, and the vessel's value of hooks for its current gear type.

For each turtle species, the vessel calculates probInteraction and conducts a Bernoulli trial. For any turtle species for which the trial is positive, the HLF variable turtleInteractionsSoFar for the turtle species is incremented.

4.3.7 Return to port decision

This submodel is conducted by vessels to decide whether, on the next day, to return to port or stay and attempt another day of fishing. The submodel is executed after a vessel has either made a fishing set or waited because fishing was not expected to be profitable on the current day (via the fishing cell selection submodel, Section 4.3.4). The decision is based on one of two alternative financial assumptions, both of which are described belowPotential turtle interactions are not considered in this decision. When the number of turtle interactions so far (HLF variable turtleInteractionsSoFar for each turtle species) is well below the quota, the probability of fishery closure due to the interaction quota is negligibly small. When turtleInteractionsSoFar is very close to the quota, vessels are assumed to continue fishing anyway to catch as much as they can before the quota is met.

A vessel's decision of whether to return to port depends on its financial state, but a vessel also may need to return to port due to fuel and supply limitations, which are checked first.

First, the vessel checks whether it must return due to its maximum trip length. If the number of days since leaving port (current date minus the date of departure on the trip) plus the number of days needed to return (vessel variable travelTime) equals or exceeds the vessel variable maxDaysInSea for its current gear type, then the vessel decides to return to port.

Second, the vessel checks whether to return due to fuel. If the vessel variable fuelRemaining is not greater than the fuel required to fish two more days and still return to port, then the vessel decides to return. (This assumption leaves one day of fuel reserve.) The fuel required to fish two more days and return to port is equal to vessel variable fuelPerDay times (travelTime + 2).

Third, the vessel checks whether it has storage for another day's fish catch. This step requires calculating the day's expected catch *C* (with separate values of *C* for each catch species). *C* is calculated simply as the average catch per set over the fishing trip so far, except that the expected catch calculated in planning the trip (*expectedCPUE* × *hooks*; Section 4.3.2) is included in this average as one day's catch. This inclusion of the expected catch calculated in trip planning allows the vessel to estimate catch on the first day of fishing, and represents how the original estimate of catch affects decisions in a way that decreases as more fishing sets are made. The current day's expected catch of each catch species is calculated as:

$$C = \frac{(expectedCPUE \times hooks) + totalFish}{1 + numberOfSets}$$

where totalFish refers to the trip total catch of the species. (ExpectedCPUE has a non-zero value only for the target species of the vessel's current gear type.) If the sum, over all catch species, of *C* plus totalFish exceeds the vessel's value of fishHoldingCapacity, then the vessel decides to return to port.

If the above three checks have not resulted in a decision to return to port, the vessel then determines whether to return to port due to financial reasons. FMMHLF includes two alternative methods for making this decision; these implement two alternative assumptions about financial decision-making and are described in the following subsections.

4.3.7.1 Expected net revenue method

This method is used if the vessel's value of revenueTarget is less than or equal to zero. It assumes that vessel operators base their return to port decision on profit maximization: a vessel plans to continue fishing if its expected net revenue from continuing to fish another day is positive. Expected net revenue is equal to expected gross revenue minus two costs: a day's operating cost and any cost due to fish in storage becoming too old to sell. The expected gross revenue is equal to the sum, over all catch species, of *C* times expected price. Expected price is calculated as in the fish sale submodel (Section 4.3.7.2) using the current day's value of priceToday.

The operating cost for another fishing set is simply the vessel variable operatingCost.

The cost of fish in storage becoming too old to sell is assumed equal to the value of fish that, on the next day, would have their storage age at marketing (the number of days since caught, plus the time to travel to port and market the fish) exceed the maximum saleable storage time. The maximum saleable storage time is equal to the HLF parameter maxStorageDays (d), which has separate values for each gear type (60 days for SWF gear and 18 days for TUNA gear). The weight of fish that will become, on the next day, too old to sell is equal to the catch made *n* days ago, where *n* is maxStorageDays minus travelTime (which considers that vessels market fish one day after arriving in port). The value of such fish is the sum, over all the catch species, of their weight times their expected price, calculated as in the fish sale submodel (Section 4.3.7.2) using the current day's value of priceToday.

This submodel was explored by contour-plotting expected net revenue for another day of fishing, for ranges of values in the trip-planning expectation of catch and the catch actually experienced in the trip (Figure 6).

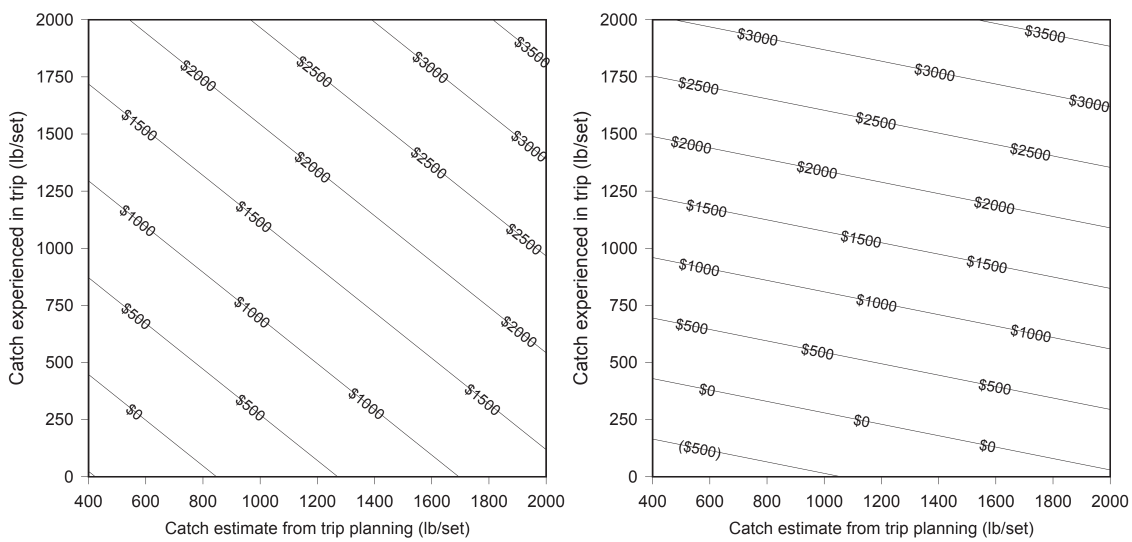


Figure 6. Exploration of the return to port submodel: plots of how expected net revenue for an additional day of fishing varies with (X) the estimated daily catch made in trip planning and (Y) the average catch experienced during the fishing trip. Assumptions include daily operating cost = \$1000 and fish price of \$2.36/lb. Left panel: expected net revenue after 1 day of fishing, which depends equally on the expected and experienced catch. Right panel: expected net revenue after 4 days of fishing, which varies much more with experienced catch.

4.3.7.2 Gross revenue target method

This method is used if the vessel's value of revenueTarget is greater than zero. It assumes that vessel operators use a less-rational decision method, returning to port only when the gross value of their catch meets a fixed target. This assumption can lead to behaviors different than the expected net revenue method in two interesting ways. First, if a vessel experiences a high catch it will return to port sooner (as soon as its target is reached) instead of continuing to exploit the catch. But second, if a vessel experiences low catch it will continue fishing even if losing money.

This method is simple. Gross revenue so far is calculated as the total weight of fish already caught that has not exceeded its maximum saleable storage time (as explained in the previous subsection), times its expected price (also as calculated in Section 4.3.7.1). (Revenue is of course calculated separately for each catch species and summed over them.) If and only if this gross revenue so far is equal to or greater than the vessel's value of revenueTarget, the vessel decides to return to port.

4.3.8 Fish sale

Vessels use this submodel to determine the value of their catch upon arrival at port after a fishing trip. The submodel is designed so that the price vessels receive can vary with daily market conditions (represented by the HLF variable priceToday) and also differ among vessels (to reflect vessel equipment, skill, and reputation for handling fish). The price also depends on catch species and gear type (so that, for example, tuna caught as bycatch during a SWF set can have a different price than tuna catch with tuna gear).

The price ($/lb) a vessel receives for each combination of catch species and gear type is calculated as:

$$fishPrice = (priceAdjFactor + priceConstant + priceMonthFactor + priceYearFactor) + (priceMultiplier \times priceToday)$$

where priceAdjFactor is a vessel variable with separate values for each catch species and gear type; priceConstant is a HLF variable with separate values for each catch species and gear type; priceMonthFactor is a HLF variable with separate values for each month, catch species, and gear type; and priceYearFactor is a HLF variable with separate values for each value of baseYear, catch species, and gear type. PriceMultiplier is an HLF variable with separate values for each combination of gear type and catch species.

The value of fishPrice is limited to make sure the parameters do not produce unreasonable values. If the value of fishPrice calculate by the above equation is negative, it is set to 0.2 × priceToday. If fishPrice is > 3 × priceToday, it is set to 3 × priceToday.

The vessel's gross revenue from a trip is then calculated by summing, over all catch species, the product of fishPrice and the vessel variable totalFish (lbs), for the gear type it used on the trip. However, any fish that were caught more than maxStorageDays (a parameter of the HLF; Section 2.2.1) are assumed to have no value.

APPENDIX A. DEVELOPMENT OF PARAMETER VALUES OF HLF

Due to the confidentiality of the data that were used in developing the parameter values of HLF, we could only provide a general description of the methods and data sources that were used here.

A1. Data Sources

Three datasets were used in deriving all the parameters that initialize the simulation model.: HLF logbook data, UFA auction data, and fishery observer data.

- **HLF Logbook data**: Hawaiian longline fisheries is mandated by the federal regulations to submit the logbook report to NMFS and report the daily fishing activities after each fishing trip. Samples of current logbook forms are available at (http://www.pifsc.noaa.gov/fmb/fmap /federal_forms/index.php.) Logbook data has information of each fishing set that was made by HLF such as the vessel that made the fishing set, the date and time a fishing set was made, the date and time a fishing set was hauled, the location of a fishing set (longitude, and latitude), the number of hooks of a fishing set, and the catches in terms of number of different fish species from a fishing set. The logbook data, however, does not contain information of the weight or size of the catches. The logbook data is primarily used to derive the fishing-related input parameters in the model such as CPUE. In this application, the HLF logbook data of the period 2006-2009 is used.
- **UFA Auction data**: Fish caught in the Hawaii-based longline fisheries and landed in Honolulu predominantly is sold in the United Fishing Agency (UFA) fish auction. The UFA provided auction data for each transaction (vessel, species, weight and price sold). The data however does not contain information re the size of fish. The UFA auction data is used to derive the input parameters that characterize the HLF fish market, i.e., daily fish price. In this application, the UFA auction data of the period 2006-2009 is used.
- **HLF Observer Data**: The HLF has been monitored under a mandatory observer program, i.e., the Pacific Islands Observer Program (PIOP). Vessels are required to carry observers, when directed to do so by the NMFS to document the incidental capture of sea turtles. The data are used to verify turtle takes as well as seabird and marine mammal interactions in the fishery. Since 2006, all swordfish fishing trips were covered by the observer program. Tuna fishing trips were randomly selected and assigned with observers. The observer data is used to derive the turtle interaction parameters, i.e., IPUE. In this application, the observer data of the period 2001-2007 is used.

A2. Methods

We conducted statistical analysis in STATA to derive the input parameters (CPUE, IPUE, Fish price) that needed for model initiations, using the three datasets described above. The specific methods that were used are as below.

A2.1. CPUE

CPUE is catch (weight) per 1,000 hooks. As HLF logbook data has only catches in terms of # of fish, we have to estimate the average fish weight from the UFA auction data. From UFA auction data, the average fish weight from each fishing trip was calculated. Average fish weight= total fish weight/total # of fish sold. This average weight was used to calculate the fish catch (weight) from each fishing set in a fishing trip. A linear regression model then was employed to derive the mean CPUE of a fishing cell, with the consideration of seasonal effects. The monthly effect is represented by 11 dummy variables (Dm, m=1 to 11). The yearly effect is represented by 3 dummy variables (Dy, y=1 to 3). The grid effect is represented by 994 dummy variables (Dg, g=1 to 994). The vessel effect is represented by 139 dummy variables(Dv, v=1 to 139).

$$CPUE(i) = a(i) + Dm \times Month + Dy \times Year + Dv \times Vessel + Dg \times Grid$$

From the above regression model, we obtained the parameters that are used for vessel to predict the fish catch (Sections 4.3.1, 4.3.5). The relation between the estimated coefficients from the regression and the model variables are described as follows.

CPUEAdjFactor = Dv
gridCellCPUE=Dg
monthlyCPUE=Dm
yearlyCPUE=Dy

A2.2 IPUE

IPUE is interaction per 1,000 hooks. HLF observer data provided the turtle interaction data (species, date, injury type, etc). The total fishing effort (# hooks) is calculated from the HLF logbook data. A linear regression model then was employed to derive the mean IPUE of a fishing grid at 4*4 degree level, with the consideration of seasonal effects. The monthly effect is represented by 11 dummy variables (Dm, m=1 to 11). The yearly effect is represented by 3 dummy variables (Dy, y=1 to 3). The grid effect is represented by 98 dummy variables (Dg, g=1 to 98).

$$IPUE(i) = a(i) + Dm \times month + Dy \times Year + Dg \times Grid$$

From the above regression model, we obtained the parameters that are used for vessel to predict the turtle interaction (Section 4.3.1). The relation between the estimated coefficients from the regression and the model variables are described as follows.

gridCellIPUE=Dg
monthlyIPUE=Dm
yearlyIPUE=Dy

A2.3 Fish Price

UFA auction data of the period 2006-2009 is used to derive the daily mean fish price. A linear regression model was employed to estimate the mean market price of a fish species, with the consideration of seasonal effects and price premium commanded by individual vessels. The monthly effect is represented by 11 dummy variables (Dm, m=1 to 11). The yearly effect is represented by 3 dummy variables (Dy, y=1 to 3). The vessel effect is represented by 139 dummy variables (Dv, v=1 to 139).

$$Price(i) = a(i) + Dm \times month + Dy \times Year + Dv \times Vessel$$

From the above regression model, we obtained the parameters that are used for vessel to predict the fish price and value (Section 4.3.8). The relation between the estimated coefficients from the regression and the model variables are described as follows.

priceConstant=a(i)
priceMonthFactor=Dm
priceYearFactor=Dy
priceAdjFactor= Dv

AGENT-BASED FISHERY MANAGEMENT MODEL OF HAWAII'S LONGLINE FISHERIES (FMMHLF)

B. Software Guide

Table of Contents

1 INTRODUCTION 35
2 EXECUTING THE MODEL 35
2.1 Running a Single Simulation 35
2.2 Conducting a Parameter Sweep 38
2.2.1 Parameter sweep directly over a numeric parameter 38
2.2.2 Parameter sweep over input files 40
3 GENERAL INFORMATION ON INPUT FILES 42
3.1 File Formats 42
3.2 Checking Input Consistency 43
4 INPUT FILE CONTENTS 43
4.1 Catch Species 43
4.2 Gear Types 44
4.3 Turtle Species 44
4.4 Geographic Information 45
4.5 Ocean Regions 46
4.6 Catch Regulations 46
4.7 Fishing Effort Regulations 46
4.8 Area Closures 46
4.9 Turtle Interaction Regulations 47
4.10 Catch Per Unit Effort (CPUE) Coefficients 47
4.11 Turtle Interaction Per Unit Effort (IPUE) Coefficients 48
4.12 Price Coefficients 48
4.13 Daily Fish Price Coefficients 49
4.14 Vessel Characteristics 49
4.15 Vessel Memory 51
5 CREATING AND EDITING THE AREA CLOSURE FILE 52
5.1 Overview 52
5.2 How to Create "Closures.kml" with Google Earth 52
5.3 How to Edit or Revise Closures.kml 59
6 ACKNOWLEDGEMENTS 60

Input Variable Index

Area 12
AverageMultiplier 16
Catch Quota 13
Catch Quota Timeline 13
CatchSpeciesCode 10, 13, 15, 17
CatchSpeciesName 10
Cell id 12
Coefficient 15
CoefficientName 15, 16
ConstantPrice 16
CPUE_SWF_BYT 18
Crew 18
EffortQuota 13
EffortQuotaTimeline 13
Fishcapacity 17
FixedCost 18
FuelCapacity 17
FuelPerHour 17
GearTypeCode 10, 11, 13, 15
GearTypeDescription 11
GearTypeName 11
HookSWF 18
HookTUNA 18
IPUE_LOG 19
IsHome 12
Latitude 12, 15, 20
Longitude 12, 15, 20
MonthlyCPUE 15
MonthlyPrice 16
Network 18
OceanRegionCode 12, 13
OceanRegionDescription 13
OceanRegionName 13
OperatingCostSWF 18
OperatingCostTUNA 18
OwnerEthnicity 18
OwnerOperated 18
PRICE_TUNA_BYT 19
PrimaryCatchSpeciesCode 11
RegulatedGearTypeCodes 14
RevenueShare 18
RevenueTarget 17
Size 17
Speed 17

TemporalIndex 15, 16
Timeline 14
TravelCost 18
Turtle Cap 14
TurtleSpeciesCode 11, 14
TurtleSpeciesDescription 11
TurtleSpeciesName 11
Type 17
Vessel 17, 20
WagePerDay 18
YearlyCPUE 15
YearlyPrice 16

1 INTRODUCTION

The purpose of this document is to describe how to use the software for Version 3.0 of the Fishery Management Model of Hawaii's Longline Fisheries (FMMHLF), developed by the Department of Natural Resources and Environmental Management of the University of Hawaii under a NSF grant "An exploratory application of agent-based modeling for policy evaluations in Hawaii's longline fishery" (SES-0918185). The model's software operates in the AnyLogic platform and uses a number of input files to describe the fishery and initial conditions.

This document uses many terms (e.g., by-catch; catch species; gear type) that are defined explicitly for this model in the separate model description document.

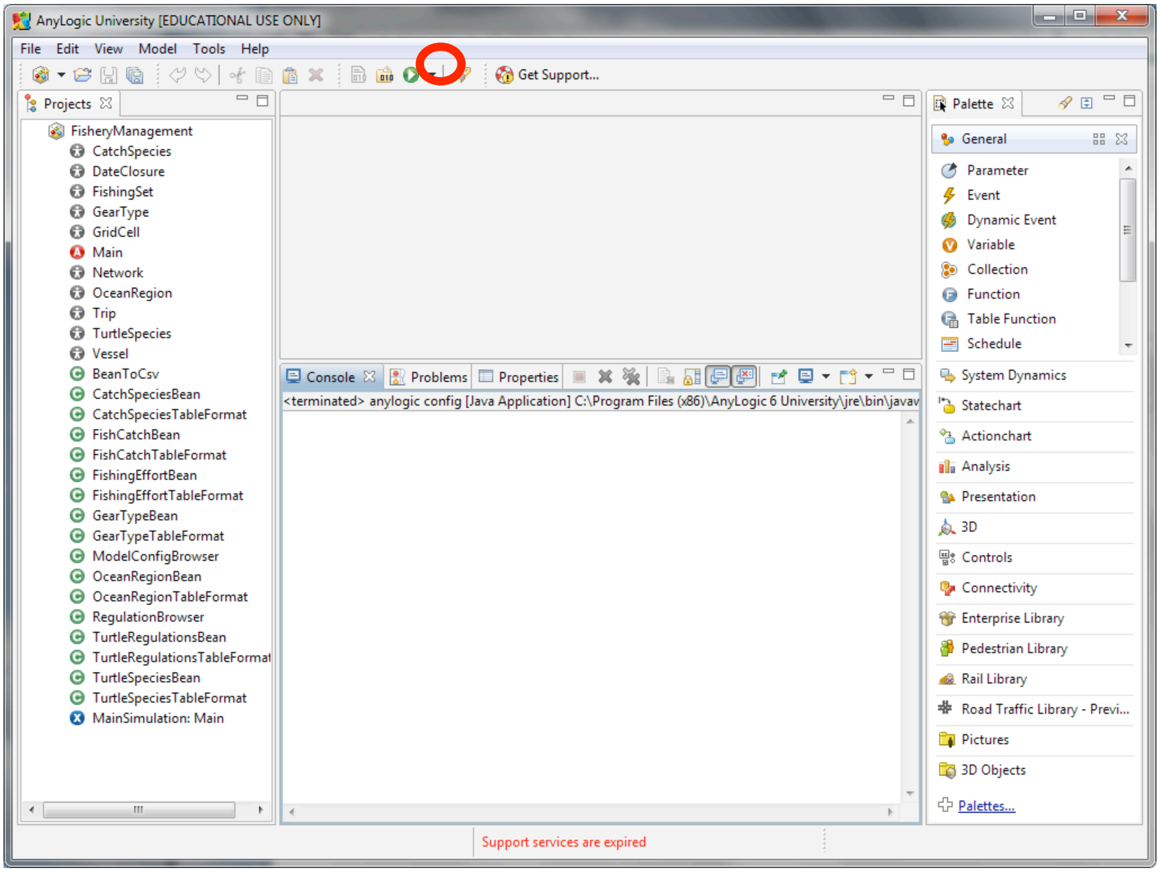
Section 2 of this document provides basic information on executing the model, and detailed instructions for setting up and executing automated simulation experiments given the input of different model parameters ("parameter sweeps"). General information and details of input to the model are provided in sections 3 and 4. Finally, Section 5 provides instructions for generating input to define a fishery area closure regulation.

2 EXECUTING THE MODEL

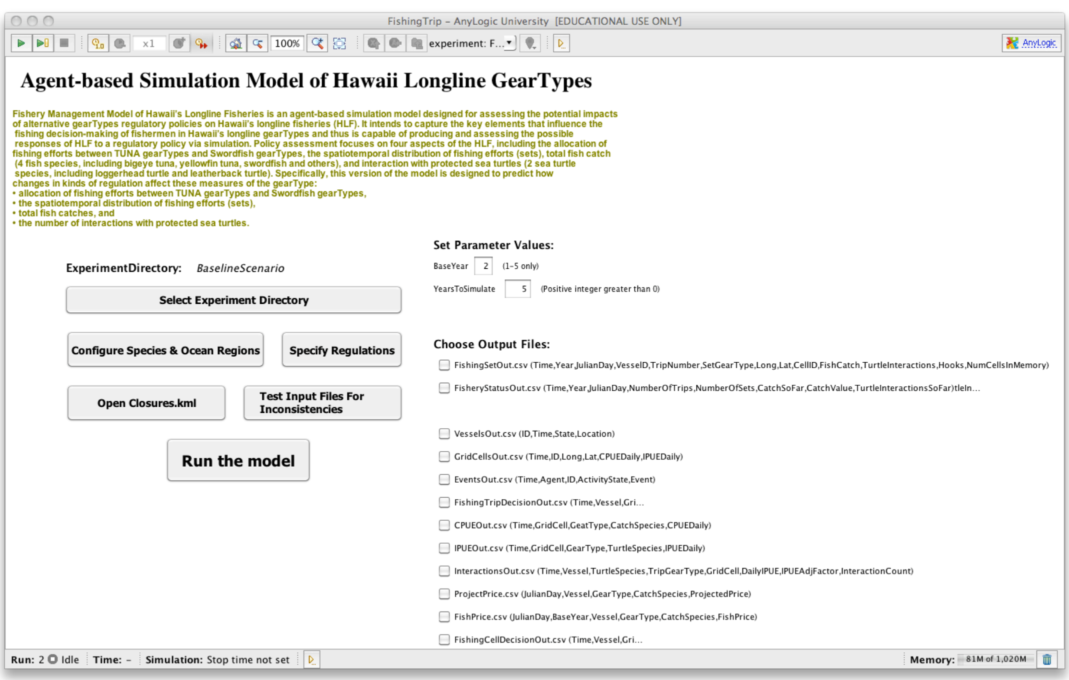
2.1 Running a Single Simulation

FMMHLF operates inside AnyLogic, so can only be used on computers where AnyLogic (which is proprietary and requires a license) is installed. The model is normally distributed as a zip archive file that, when un-zipped, creates a directory that contains an AnyLogic file (e.g., FisheryManagement.alp) and at least one subdirectory containing a set of input files. Double-clicking on the AnyLogic file starts up AnyLogic and loads the code.

Within AnyLogic, the model is started by clicking on the small green "run" button at the top of the window. (If this opens a dialog instead of immediately starting the model, select something like "Run MainSimulation".)



When the model starts, it opens a graphical user interface illustrated here:

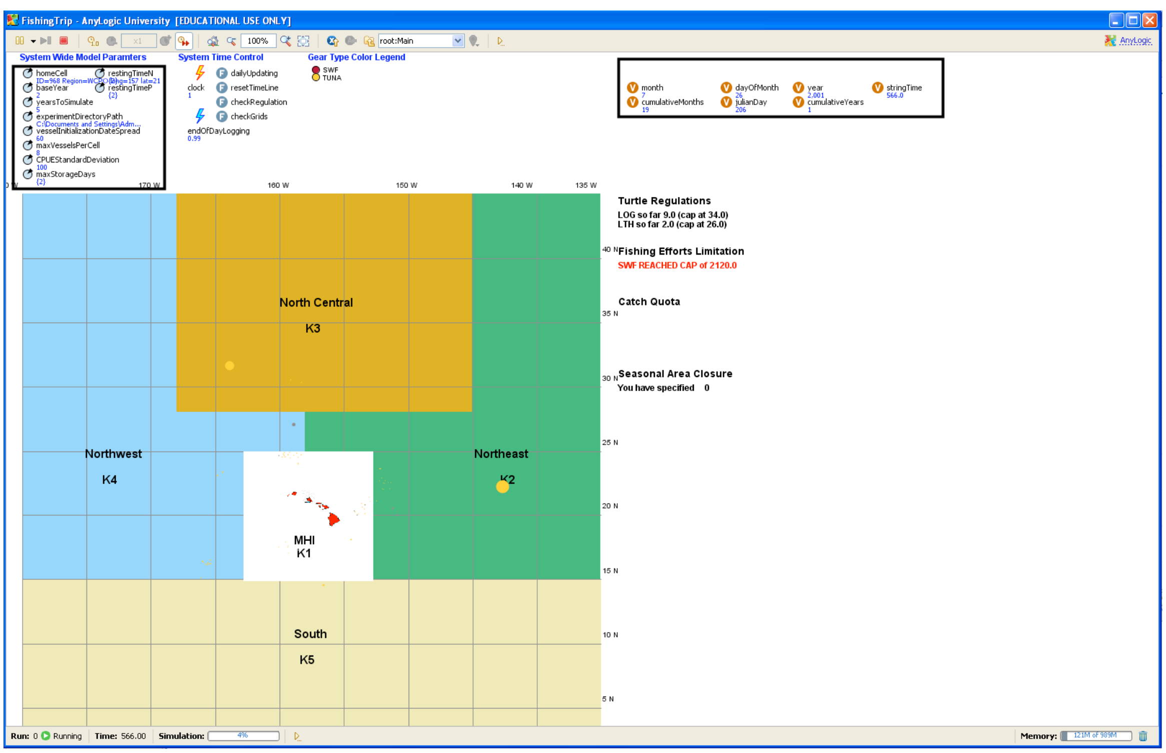


The buttons on the left side of the screen allows you to do the following things.

- "Select Experiment Directory" lets you select or browse to the directory containing the input files you want to use. (The code automatically looks for subdirectories to where the code resides as alternative experiment directories.)
- "Configure Species and Ocean Regions" opens a dialog in which you can edit the information on fish species, gear types, ocean regions, and turtle species that in the input files described below at sections 4.1, 4.2, 4.3, and 4.5. (The dialog actually edits these files.)
- "Specify Regulations" lets you edit the information on quotas described in sections 4.6, 4.7, and 4.9.
- "Open Closures.kml" launches Google Earth software with the current area closures so you can edit the area closed to fishing (or the dates and species the closure applies to), as described in Section 5.
- "Test Input Files For Inconsistencies" is described in Section 3.2.
- "Run the model" actually starts model execution.

You can type numbers into the boxes under "Set Parameter Values" to change the base year from the input data used in the simulation or the length of the run. Finally, check or uncheck the boxes next to the output files to specify what kinds of data should be logged during the run.

When the "Run the model" button is clicked, the model's graphical display appears:



This display shows the grid cells, vessels, and information on simulation status. Buttons on the upper left can be used to pause, resume, or stop execution. Mouse clicks on grid cells or vessels open up a display to their state; to return to the full model display after doing so, click on the red button labeled "Return to root object".

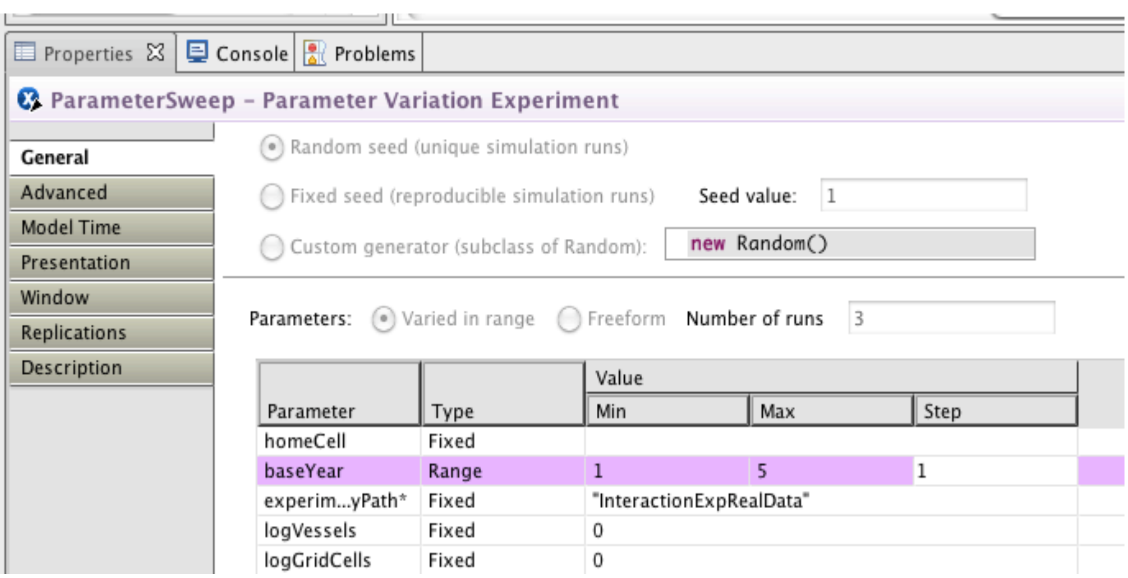
2.2 Conducting a Parameter Sweep

A "parameter sweep" is the process of running the model multiple times while systematically varying one parameter from run to run. For example, you may be interested to see the variation in model outputs that result from changing the "baseYear" parameter. Alternatively, you may be interested in varying the limit on turtle interactions over a range of values to assess the impact of management strategies on fish catch. Both of these model runs are illustrated as examples in detail below.

2.2.1 Parameter sweep directly over a numeric parameter

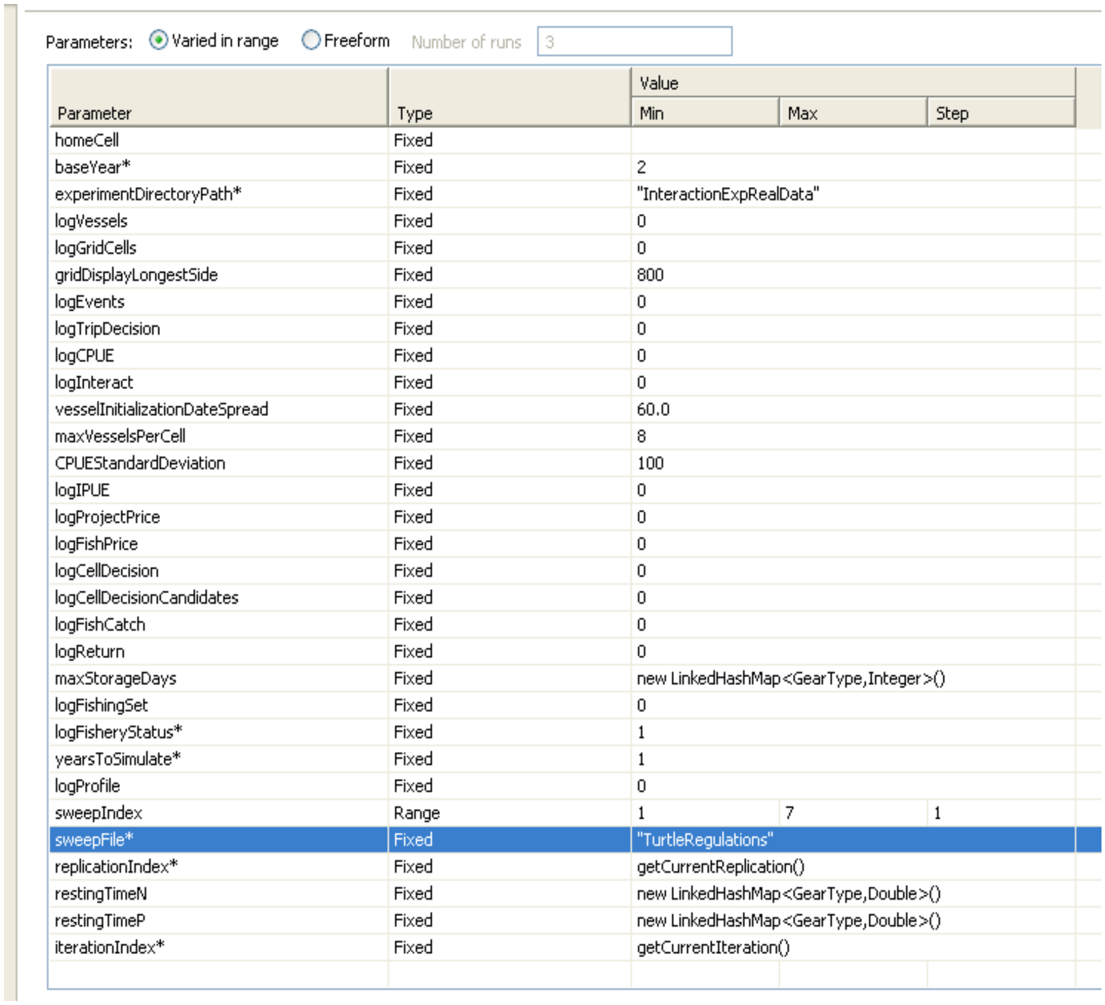
This kind of parameter sweep varies a parameter that is a number.

- To conduct a parameter sweep, open the "ParametersVariation" entity under the FisheryManagement project in the "Projects" window.
- Look in the "Properties" pane >> "General" tab to see a table of available parameters to automatically vary in the experiment. Only a handful of parameters can be varied directly using this table:
  - baseYear
  - vesselInitializationDateSpread
  - maxVesselsPerCell
  - CPUEStandardDeviation
  - yearsToSimulate
- As an example, we will vary baseYear. Change the Type in the table from "Fixed" to "Range", then under the "Value" column there should be 3 sub-cells. Type the min and max values and the step size over which you want the parameter to vary. For example, in the illustration below, we've added 1, 5, and 1 to the sub-cells, which will cause the model to be executed for all integer values of baseYear from 1 to 5.



- All other parameters should have their Type set to "Fixed". To enable logging of results to output files, set the value next to any log file you want produced (especially, "logFisheryStatus") to "1" (a value of "0" turns logging off).

- Make sure the parameter “experimentDirectoryPath” has the name of the folder containing the input files you want to use in the parameter sweep.
- Set sweepFile to “none” (including the quotes). This signals the experiment manager that you are sweeping directly over a parameter and not over an input file.



Parameters: Varied in range Freeform Number of runs 3

Parameter	Type	Value Min	Max	Step
homeCell	Fixed			
baseYear*	Fixed	2		
experimentDirectoryPath*	Fixed	"InteractionExpRealData"		
logVessels	Fixed	0		
logGridCells	Fixed	0		
gridDisplayLongestSide	Fixed	800		
logEvents	Fixed	0		
logTripDecision	Fixed	0		
logCPUE	Fixed	0		
logInteract	Fixed	0		
vesselInitializationDateSpread	Fixed	60.0		
maxVesselsPerCell	Fixed	8		
CPUEStandardDeviation	Fixed	100		
logIPUE	Fixed	0		
logProjectPrice	Fixed	0		
logFishPrice	Fixed	0		
logCellDecision	Fixed	0		
logCellDecisionCandidates	Fixed	0		
logFishCatch	Fixed	0		
logReturn	Fixed	0		
maxStorageDays	Fixed	new LinkedHashMap<GearType,Integer>()		
logFishingSet	Fixed	0		
logFisheryStatus*	Fixed	1		
yearsToSimulate*	Fixed	1		
logProfile	Fixed	0		
sweepIndex	Range	1	7	1
sweepFile*	Fixed	"TurtleRegulations"		
replicationIndex*	Fixed	getCurrentReplication()		
restingTimeN	Fixed	new LinkedHashMap<GearType,Double>()		
restingTimeP	Fixed	new LinkedHashMap<GearType,Double>()		
iterationIndex*	Fixed	getCurrentIteration()		

- Increase the number of replications as desired. Replication means that the model is run multiple times with the same set of inputs, with only the random numbers differing among replicates of each parameter value. During a parameter sweep, the “iterations” refer to the number of unique values of the parameter being varied. For each of those iterations, you may conduct as many replications as you like.

  Under the “Properties” pane >> “Replications” tab, make sure that “Use Replications” is checked. Set the value in the box labeled “Replications per iteration” to the number of replications you want. **If you do not want replications, set the number of replications to 1 instead of unchecking the box labeled “Use Replications.”**
- Finally, run the parameter sweep experiment by selecting the menu item “Model” >> “Run” >> “ParametersVariation / ParameterSweep” then click the “Run Experiment” button.
- If you enabled any of the log files, then as the experiment runs these files will be created with the same name as during a single simulation run, but the iteration and replication numbers

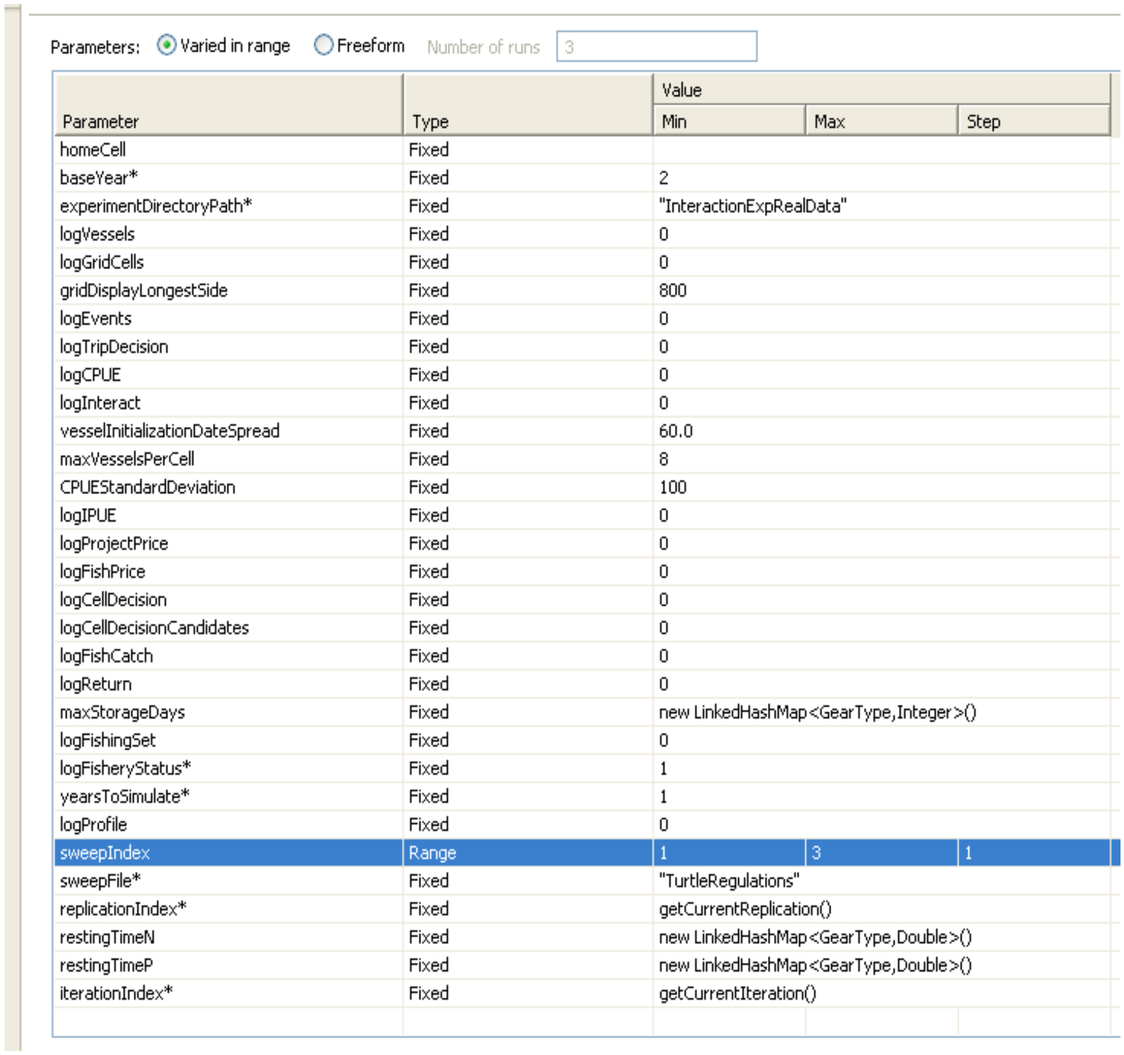
will be appended to the filenames. For example, if we enabled logFisheryStatus in the above example and set the number of replications to 2, we would get the following output files (note that the iteration is the first appended number followed by an underscore and then the replication):

- FisheryStatusOut1_1.csv
- FisheryStatusOut1_2.csv
- FisheryStatusOut2_1.csv
- FisheryStatusOut2_2.csv
- FisheryStatusOut3_1.csv
- FisheryStatusOut3_2.csv
- FisheryStatusOut4_1.csv
- FisheryStatusOut4_2.csv
- FisheryStatusOut5_1.csv
- FisheryStatusOut5_2.csv

2.2.2 Parameter sweep over input files

For this model, most parameters are in input files, so parameter sweeps involve varying the data contained in one of the input files. This is done by creating separate copies of the input file that each contain one set of parameter values, then running the model using each such input file. For example, we will conduct a parameter sweep that varies the quota on turtle interactions contained in the file "TurtleRegulations.csv".

- Find the file "TurtleRegulations.csv" and make copies of the file, one for each value of the quota you wish to simulate. Name the files "TurtleRegulations#.csv" but replace "#" with increasing integer numbers beginning at "1". For example, to test three different interaction quotas we create three copies of TurtleRegulations.csv and name them as follows:
    - TurtleRegulations1.csv
    - TurtleRegulations2.csv
    - TurtleRegulations3.csv
- Then edit the regulation files to contain the interaction quotas you wish to simulate.
- In Anylogic, open the "ParametersVariation" entity under the FisheryManagement project in the "Projects" window.
- Look in the "Properties" pane, then "General" tab to see the table of parameters.
- Set the Type of "sweepIndex" to "Range" and configure it to vary over the same index values used in the file names. In this example we set it to vary from 1 to 3 by steps of 1. Then set the "sweepFile" parameter to be the name of the input file you whish to vary. Exclude the file extension and surround the name in quotes. For this example we use "TurtleRegulations".



Parameters: (•) Varied in range ( ) Freeform Number of runs 3

Parameter	Type	Value		
		Min	Max	Step
homeCell	Fixed			
baseYear*	Fixed	2		
experimentDirectoryPath*	Fixed	"InteractionExpRealData"		
logVessels	Fixed	0		
logGridCells	Fixed	0		
gridDisplayLongestSide	Fixed	800		
logEvents	Fixed	0		
logTripDecision	Fixed	0		
logCPUE	Fixed	0		
logInteract	Fixed	0		
vesselInitializationDateSpread	Fixed	60.0		
maxVesselsPerCell	Fixed	8		
CPUEStandardDeviation	Fixed	100		
logIPUE	Fixed	0		
logProjectPrice	Fixed	0		
logFishPrice	Fixed	0		
logCellDecision	Fixed	0		
logCellDecisionCandidates	Fixed	0		
logFishCatch	Fixed	0		
logReturn	Fixed	0		
maxStorageDays	Fixed	new LinkedHashMap<GearType,Integer>()		
logFishingSet	Fixed	0		
logFisheryStatus*	Fixed	1		
yearsToSimulate*	Fixed	1		
logProfile	Fixed	0		
sweepIndex	Range	1	3	1
sweepFile*	Fixed	"TurtleRegulations"		
replicationIndex*	Fixed	getCurrentReplication()		
restingTimeN	Fixed	new LinkedHashMap<GearType,Double>()		
restingTimeP	Fixed	new LinkedHashMap<GearType,Double>()		
iterationIndex*	Fixed	getCurrentIteration()		

- All other parameters should have their “Type” set to “Fixed” and contain appropriate values. To enable logging of output to files, set the value next to any log file (e.g., “logFisheryStatus”) to a “1” (a value of “0” turns logging off).
- Make sure the parameter “experimentDirectoryPath” has the name of the folder containing the input files you want to use in the parameter sweep.
- Increase number of replications if desired. Under the “Properties” pane >> “Replications” tab, make sure that “Use Replications” is checked. Set the value in the box labeled “Replications per iteration” to the number of replications you want. A replication means that the model is run multiple times with the same set of inputs. During a parameter sweep, the “iterations” refer to the number of unique values of the parameter being varied. For each of those iterations, you may conduct as many replications as you like. **Note: If you do not want replications, set the number of replications to 1 instead of unchecking the box labeled “Use Replications.”**
- Finally, run the parameter sweep experiment by selecting the menu item “Model” >> “Run” >> “ParametersVariation / ParameterSweep” then click the “Run Experiment” button.
- If your parameter sweep experiment enabled any of the log files, then while the experiment runs, these files will be created with the same name as during a single simulation run, but the

iteration and replication numbers will be appended to the filenames. For example, if we enabled logFisheryStatus in this example and set the number of replications to 2, we would get the following output files (note that the iteration is the first appended number followed by an underscore and then the replication):

- FisheryStatusOut1_1.csv
- FisheryStatusOut1_2.csv
- FisheryStatusOut2_1.csv
- FisheryStatusOut2_2.csv
- FisheryStatusOut3_1.csv
- FisheryStatusOut3_2.csv

3 GENERAL INFORMATION ON INPUT FILES

This section provides information relevant to all input files. All input files for a project or simulation experiment are provided in one data directory, which normally is a subdirectory of the main model directory. Separate directories can be used for separate projects or experiments, but all input files for a model run must be in the same directory. The content of each of input file is described in Section 4.

3.1 File Formats

The input files are all in ".csv" format, a common data table format in which values in a line of data are separated by commas. (The area closure file, Section 4.8, is an exception that does not use .csv format.) On most computers, .csv files open by default in a spreadsheet (Windows opens them in Excel) but they can also be opened in text editors and word processors. (An example is in Section 4.1.)

The most convenient way to edit .csv files is typically to open them in a spreadsheet, and then make sure they are saved in .csv format after editing. Occasionally, editing a .csv input file in spreadsheet software can result in extra blank columns (extra commas at the end of each line) or rows at the end of the file, which keep the model from reading it correctly. If the model cannot open an input file after it was edited, open the .csv file in a text editor (or Word) and delete any extra commas or lines.

FMMHLF provides menus for editing some inputs; these are described below in the section for the relevant inputs and in the section on the model user interface. Other inputs must be generated and managed by editing the .csv files.

The standard format for FMMHLF input files is for the first line to contain header information, with input on all following lines. The header information includes only the names of the values in the following lines.

In general, file names and variable values are **case sensitive**, so care must be used with capitalization. For example, the model software will treat the fish species code "SWF" as different from "swf" and from "SwF".

3.2 Checking Input Consistency

The input files to FMMHLF are highly independent: the values that are valid in one file (e.g., which fish species each gear type targets; Section 4.2) often depend on values in other files (e.g., the catch species; Section 4.1). Any major changes in fish species or gear types require changes throughout the input and careful checking.

To help with such changes, a testing tool is built into the FMMHLF software. On the model interface is a button labeled "Test input files for inconsistencies". Clicking on this button executes the tool, which tells the user whether it finds any problems with how fish species, turtle species, gear types, ocean regions, and regulations are defined in the input files.

4 INPUT FILE CONTENTS

4.1 Catch Species

This file must be named "CatchSpecies.csv". It defines the species of fish that vessels can catch, including species (or categories that are treated as a species) that are by-catch. The file typically has these contents:

CatchSpeciesCode	CatchSpeciesName	CatchSpeciesDescription	GearTypeCode
BYT	Bigeye Tuna	Bigeye Tuna	TUNA
YFT	Yellowfin Tuna	YellowfinTuna	TUNA
SWF	Swordfish	Swordfish	SWF
OTHER	Other	Other fish	

CatchSpeciesCode is a text word (character string, with no spaces) with unique values for each catch species.

CatchSpeciesName is a text string containing the common name of the species.

CatchSpeciesDescription contains text (not used by the model) that can document exactly what fish are referred to.

GearTypeCode is the code for the fishing gear type that targets the species. These codes are defined in the gear type input (Section 4.2). Species that are always by-catch have no value for this code.

To illustrate the .csv file format, this input file looks like the following when opened in a text editor:

CatchSpeciesCode,CatchSpeciesName,CatchSpeciesDescription,GearTypeCode
BYT,Bigeye Tuna,Bigeye Tuna,TUNA
YFT,Yellowfin Tuna,YellowfinTuna,TUNA
SWF,Swordfish,Swordfish,SWF
OTHER,Other,Other fish,

When opened in a spreadsheet, the file looks like this:

	A	B	C	D
1	CatchSpeciesCode	CatchSpeciesName	CatchSpeciesDescription	GearTypeCode
2	BYT	Bigeye Tuna	Bigeye Tuna	TUNA
3	YFT	Yellowfin Tuna	YellowfinTuna	TUNA
4	SWF	Swordfish	Swordfish	SWF
5	OTHER	Other	Other fish	
6				

4.2 Gear Types

This file is named "GearTypes.csv" and defines the types of fishing gear that vessels can have. The file typically has these contents:

GearTypeCode	GearTypeName	GearTypeDescription	PrimaryCatchSpeciesCode
SWF	Swordfish	Swordfish	SWF
TUNA	Tuna	The primary species within Tuna is Bigeye Tuna	BYT

GearTypeCode is a text word with a unique value for each gear type.

GearTypeName is a text string defining the name of the gear type; these typically refer to the fish species targetted by the gear.

GearTypeDescription is descriptive text not used by the computer.

PrimaryCatchSpeciesCode is the code (Section 4.1) for the catch species that is the primary target of the gear type. Note that the catch species input (Section 4.1) allows multiple species of fish to be caught by the same gear type, even though this gear type input assigns only one catch species as the primary target.

4.3 Turtle Species

This file is named "TurtleSpecies.csv" and defines the types of species of turtles for which the model has regulations. The file typically has these contents:

TurtleSpeciesCode	TurtleSpeciesName	TurtleSpeciesDescription
LOG	Loggerhead	Loggerhead turtle
LTH	Leatherback	Leatherback turtle

TurtleSpeciesCode is a text word providing a unique code for each turtle species.

TurtleSpeciesName is the common name of the turtle species.

TurtleSpeciesDescription provides a text description of the species, not used by the computer.

4.4 Geographic Information

This file is named "GeoInfo.csv". It defines the spatial information used by the model, with one row for each grid cell. Because the file provides input for combinations of gear types and catch species and turtle species, the number of columns in the file depends on the values input in the gear type, catch species, and turtle species files (Sections 4.1, 4.2, 4.3).

The input provided for each cell in a typical FMMHLF run is defined in the following table; the first column in the table contains the variable names as they appear in the first row of the input file. (The following table does *not* look like the input file!)

Variable	Definition	Value type	Typical values
id	An identification number unique to each cell.	Integer	1, 2, 3...
Longitude	Longitude (degrees) of the cell center	Integer	135, 136, 137...
Latitude	Latitude (degrees) of the cell center	Integer	0,1,2...
Area	A code for the geographic area the cell belongs to. This code is not used.	Integer	1, 2,...5
OceanRegionCode	Code for the ocean region the cell is in (Section 4.5)	Text word	EP, WCPO
IsHome	Whether the cell is the fishing fleet's home port	0 or 1	1 = Cell is home port; only one cell can be home
SWF_BYT	The cell-specific variable in the equation for CPUE (*gridCellCPUE* in the model description), for catch species BYT when fishing with gear type SWF	float	-6.474, 111.901, -53.003... Values of -9999 indicate CPUE of zero in cell.
SWF_YFT	Same as above, for catch species YFT when fishing with gear type SWF	"	"
SWF_SWF	(etc.)		
SWF_OTHER			
TUNA_BYT			
TUNA_YFT			
TUNA_SWF			
TUNA_OTHER			
SWF_LOG	The cell-specific variable in the equation for IPUE (*gridCellIPUE* in the model description), for turtle species LOG when fishing with gear type SWF	float	0.000199, 0.0137, 0.0085...
SWF_LTH	Same as above, for turtle species LTH when fishing with gear type SWF	"	"
TUNA_LOG	(etc.)		
TUNA_LTH			

4.5 Ocean Regions

This file is named “OceanRegions.csv” and defines the ocean regions used to represent fishing regulations (quotas that vary among regions). The file typically has these contents:

OceanRegionCode	OceanRegionName	OceanRegionDescription
WCPO	Western and Central Pacific Ocean	Western and Central Pacific Ocean
EPO	Eastern Pacific Ocean	Eastern Pacific Ocean

OceanRegionCode is a text word defining a region’s code. OceanRegionName is a text string providing the full name of the region. OceanRegionDescription is a text description of the region, ignored by the computer.

4.6 Catch Regulations

This file is named “FishCatchRegulations.csv”. It defines catch quota regulations, with one row per quota. The file typically has contents such as these:

OceanRegionCode	CatchSpeciesCode	Quota	Timeline
WCPO	BYT	1000000	12

OceanRegionCode is the code for the region (Section 4.5) the quota applies to. CatchSpeciesCode is the catch species (Section 4.1) of the quota. Quota is the regulatory quota, in lbs of fish (the variable catchQuota in the model description). Timeline defines the period (months) that the quota is evaluated over; a value of 12 means it is an annual quota, being reset at January 1 of each year.

4.7 Fishing Effort Regulations

This file is named “FishingEffortRegulations.csv”. It defines effort quota regulations (limits on the number of sets made with a gear type), with one row per quota. The file typically has contents such as these:

GearTypeCode	EffortQuota	EffortQuotaTimeline
SWF	2120	12

GearTypeCode is the gear type (Section 4.2) to which the quota applies. EffortQuota is the quota (number of fishing sets). EffortQuotaTimeline defines the period (months) that the quota is evaluated over; a value of 12 means it is an annual quota.

4.8 Area Closures

This file defines regulatory closures that prohibit fishing at certain times in certain geographic areas. The file is designed to be created using GoogleEarth software, and is in GoogleEarth’s .kml file format. Directions for creating and revising this file are at Section 5.

4.9 Turtle Interaction Regulations

The file TurtleRegulations.csv defines the regulatory limit ("cap") on the number of turtle interactions by fishing vessels. It typically has contents like:

TurtleSpeciesCode	Cap	Timeline	RegulatedGearTypeCodes
LTH	20	12	SWF
LOG	30	12	SWF

TurtleSpeciesCode is code for the turtle species (Section 4.3) that the interaction quota applies to. Cap is the quota: the number of interactions after which fishing is prohibited (the model description uses the variable name "TurtleCap"). Timeline is as for the previous two quota files. RegulatedGearTypeCodes is the code for the gear type (Section 4.2) that will no longer be allowed after the quota is met. (Additional lines could be added to the file to make turtle interaction quotas apply to more gear types.)

4.10 Catch Per Unit Effort (CPUE) Coefficients

This file is named "CPUECoefficients.csv". It provides the year type and month coefficients for fish CPUE (*monthlyCPUE* and *yearlyCPUE* in the equation for *CPUEMean* of each cell). The file must provide *yearlyCPUE* values by including one line (with one coefficient value) for each combination of (a) year types (as defined in the model description), (b) gear type (Section 4.2), and (c) catch species (Section 4.1). It must also then provide *monthlyCPUE* values by including one line for each combination of (a) month, from one to 12, (b) gear type, and (c) catch species. The following excerpt illustrates the file format.

CoefficientName	GearTypeCode	TemporalIndex	CatchSpeciesCode	Coefficient
YearlyCPUE	SWF	0	BYT	15.346
YearlyCPUE	SWF	0	YFT	15.605
YearlyCPUE	SWF	0	SWF	537.16
YearlyCPUE	SWF	0	OTHER	113.561
YearlyCPUE	SWF	1	BYT	44.121
YearlyCPUE	SWF	1	YFT	0
...				
MonthlyCPUE	SWF	1	BYT	114.908
MonthlyCPUE	SWF	1	YFT	24.969
MonthlyCPUE	SWF	1	SWF	1683.25
MonthlyCPUE	SWF	1	OTHER	24.378

CoefficientName is a text word with value of either "YearlyCPUE" or "MonthlyCPUE" to indicate whether the coefficient provided in the row is a value of *yearlyCPUE* or *monthlyCPUE*.

GearTypeCode indicates the gear type (Section 4.2) for the CPUE coefficients.

TemporalIndex indicates what year type or month the coefficient is for. If CoefficientName is "YearlyCPUE", then the value of TemporalIndex must be one of the (integer, 0-4) year types defined for the model. If CoefficientName is instead "MonthlyCPUE", then the value of TemporalIndex must be an integer from 1 to 12 to indicate the month.

CatchSpeciesCode indicates the catch species (Section 4.1) that the CPUE coefficient is for.

Coefficient is the CPUE coefficient value (a floating point number).

4.11 Turtle Interaction Per Unit Effort (IPUE) Coefficients

This file, named "GeneralIPUE.csv", provides coefficients used to calculate the probability of turtle interactions for individual grid cells. The values in the file are the variable *IPUEMonthly* for specific grid cells. (Most combinations of cell and month are missing from the current version of this file; the model sets *IPUEMonthly* to zero for combinations not in this file. All cells currently in the file have coefficient values of zero.) The file has the following contents:

GridID	Longitude	Latitude	TARGET	MONTH	LOG	LTH
215425	154	25	D	12	0.0	0.0
315231	152	31	D	8	0.0	0.0
416025	160	25	D	10	0.0	0.0
416317	163	17	D	11	0.0	0.0
516014	160	14	D	7	0.0	0.0

GridID is a cell identifier that is not used. Longitude and latitude identify the cell that the line refers to; there must be a cell in the geographic input (Section 4.4) with the same coordinate pair. (Note the inconsistencies in capitalization in these variable names.) Target is an unused code. Month is an integer for the month the coefficients apply to.

The final two columns provide the value of *IPUEMonthly* for the two species of turtle with codes LOG and LTH (Section 4.3). These values are floating point numbers.

4.12 Price Coefficients

This file is named "PriceCoefficients.csv". It provides the non-daily coefficients used in calculating fish market prices. (Prices also vary daily; the daily coefficients are in a separate file described at Section 4.13.) The file typically looks like:

CoefficientName	GearTypeCode	TemporalIndex	CatchSpeciesCode	Coefficient
AverageMultiplier	SWF	NA	BYT	0
AverageMultiplier	SWF	NA	YFT	0
AverageMultiplier	SWF	NA	SWF	0
AverageMultiplier	SWF	NA	OTHER	0
AverageMultiplier	TUNA	NA	BYT	0
AverageMultiplier	TUNA	NA	YFT	0
AverageMultiplier	TUNA	NA	SWF	0
AverageMultiplier	TUNA	NA	OTHER	0
ConstantPrice	SWF	NA	BYT	4.39265

ConstantPrice	SWF	NA	YFT	1.08743
ConstantPrice	SWF	NA	SWF	2.00341
...				
MonthlyPrice	SWF	1	BYT	-0.12337
MonthlyPrice	SWF	1	YFT	0.40302
MonthlyPrice	SWF	1	SWF	0.324
MonthlyPrice	SWF	1	OTHER	0.08964
MonthlyPrice	SWF	2	BYT	0.09368
MonthlyPrice	SWF	2	YFT	1.97104
...				

The format of this price coefficient file is similar to that of the CPUE coefficients file (Section 4.10). Valid values in the CoefficientName column are: "AverageMultiplier" (indicating values of the model variable *priceMultiplier* used in calculating fish prices); "ConstantPrice", indicating values of the model variable *priceConstant*; "MonthlyPrice", indicating values of the model variable *priceMonthFactor*; and "YearlyPrice", for values of the variable *priceYearFactor*.

Valid values of the TemporalIndex column depend on CoefficientName. For lines with CoefficientName equal to "AverageMultiplier" or "ConstantPrice", the value of TemporalIndex must be "NA". Where CoefficientName equals "MonthlyPrice", the value of TemporalIndex must be an integer (1-12) representing the month. Where CoefficientName equals "YearlyPrice", the value of TemporalIndex must be an integer (0 or higher) representing the year type.

The last column is the actual coeffient value, which is a floating point number.

4.13 Daily Fish Price Coefficients

This file, called "Price.csv", provides the daily variable *priceToday* used in the model's method for calculating market prices for fish. The file provides a daily value for each of the catch species (Section 4.1), so the number of columns in the file depends on the number of catch species. A typical file looks like:

JulianDay	BYT	YFT	SWF	OTHER
1	5.84	4.57	1.03	1.42
2	3.68	2.88	2	1.52
3	3.61	2.87	2	1.64
4	3.79	3.32	2.93	1.7

The JulianDay column is the day of the year (integers from 1 to 360). There is one additional column for each fish catch species, with the column label equal to the catch species' value of CatchSpeciesCode. The contents of these species columns are the daily values of *priceToday*.

4.14 Vessel Characteristics

The file "Vessel.csv" provides the characteristics of each simulated fishing vessel. It includes one row per vessel, with columns for each of the vessel variables described in this table. (The following table does *not* represent the input file! Rows in this table are columns in the file.) These variables correspond to vessel state variables, and vessel-specific CPUE and IPUE parameters, as described in the model description.

Variable	Definition	Value type	Typical values
Vessel	A unique identification number	Integer	1, 2, 3, ...
Type	A category variable representing what gear types the vessel can use	Single characters	"T" indicates a tuna-only vessel; "M" indicates a vessel with both tuna and swordfish gear
Size	The vessel's size category	Integer	1, 2, or 3
Fishcapacity	The number of lbs of fish the vessel can store	Integer	37857, 54606, 71277
FuelCapacity	Maximum fuel storage (lbs)	Integer	5472, 16980, 14725
FuelPerHour	Fuel use rate (lbs/day); note that this variable is mislabeled: it is a daily, not hourly, rate.	Integer	149, 213, 244
Speed	Vessel speed during travel to and from fishing grounds (statute miles/day)	Float	9.1, 10.8, 11.9
TripLengthSWF	Typical trip length (days) when using swordfish gear	Integer	0 (for vessels that do not use SWF gear), 36, 27
TripLengthTUNA	Typical trip length (days) when using tuna gear	Integer	20, 18, 24
MAXDaysInSea	The maximum days the vessel can stay away from port	Integer	36, 27, 54
RevenueTarget	Gross revenue target used in return to port decision ($). If zero or negative, return to port decision uses the net revenue method instead of the revenue target method.	Float	32275, 114919, -1
OwnerOperated	Whether the vessel is owned by its operator (not currently used)	Boolean	0 (not owner-operated) or 1
OwnerEthnicity	(Not used)	Integer	(Values are 0 until this variable is implemented)
RevenueShare	(Not used)	Integer	(Values are 0 until this variable is implemented)
HookSWF	Number of hooks (1000s) in a swordfish set	Float	0.701, 0.882, 0.794
HookTUNA	Number of hooks (1000s) in a tuna set	Float	1.882, 2.096, 1.870
Crew	(Not used)	Integer	(Values are 0 until this variable is implemented)
FixedCost	(Not used)	Integer	(Values are 0 until this variable is implemented)
WagePerDay	Crew wages ($/day for entire crew)	Float	442.7, 381.97, 467.93
OperatingCostSWF	The cost of fishing ($/day) when making swordfish sets	Float	654.19, 513.28, 542.44
OperatingCostTUNA	The cost of fishing ($/day) for tuna sets	Float	411.92, 434.04, 748.52
TravelCost	Operating cost when traveling to and from fishing grounds ($/day)	Float	348, 459.04, 565.09
NetWork	The social network (for sharing fishing information) that the vessel belongs to	Integer	1, 2, 3... (all vessels with the same value belong to the same network)

Variable	Definition	Value type	Typical values
CPUE_SWF_BYT	The variable *CPUEAdjFactor* used in modeling fish catch, using gear type = SWF for the catch species BYT	Float	-173.692, -119.413, -146.38 (A value of -9999 indicates the vessel does not use this gear type)
CPUE_SWF_YFT	The variable *CPUEAdjFactor*, for gear type = SWF and catch species YFT	Float	-8.281, 14.905, -9999 if the vessel does not use this gear type
CPUE_SWF_SWF	(etc.)		
CPUE_SWF_OTHER			
CPUE_TUNA_BYT			
CPUE_TUNA_YFT			
CPUE_TUNA_SWF			
CPUE_TUNA_OTHER			
PRICE_TUNA_BYT	The variable *priceAdjFactor* used in modeling fish price, using gear type = SWF for the catch species BYT	Float	0.32343, -0.1933, -0.37696
PRICE_TUNA_YFT	(etc.)		
PRICE_TUNA_SWF			
PRICE_TUNA_OTHER			
PRICE_SWF_BYT			
PRICE_SWF_YFT			
PRICE_SWF_SWF			
PRICE_SWF_OTHER			
IPUE_LOG	The variable *IPUEAdjFactor* used in modeling turtle interactions, for turtle species = LOG	Float	1.0 for all vessels because vessel-specific interaction rates have not been estimated
IPUE_LTH	*IPUEAdjFactor* for turtle species = LTH	Float	1.0 for all vessels

4.15 Vessel Memory

The input file "VesselMemory.csv" contains the initial memory of fishing grid cells for each vessel, used when the model is initialized to create the memory of social networks of vessels. The file is simply a list of cells, identified by the latitude and longitude of their center, that each vessel is given memory of. The file looks like:

Vessel	latitude	longitude
1	17	161
1	18	155
1	18	157
1	18	159
1	18	160
...		

The first column identifies the vessel, using the "Vessel" identifier from the vessel characteristics file (Section 4.14). The second and third columns contain the latitude and longitude of one cell in the vessel's memory; there are many lines per vessel to represent all the cells they have memory of. For each latitude and longitude pair in this file, there must be a cell in the geographic information file (Section 4.4) with the same coordinates.

5 CREATING AND EDITING THE AREA CLOSURE FILE

5.1 Overview

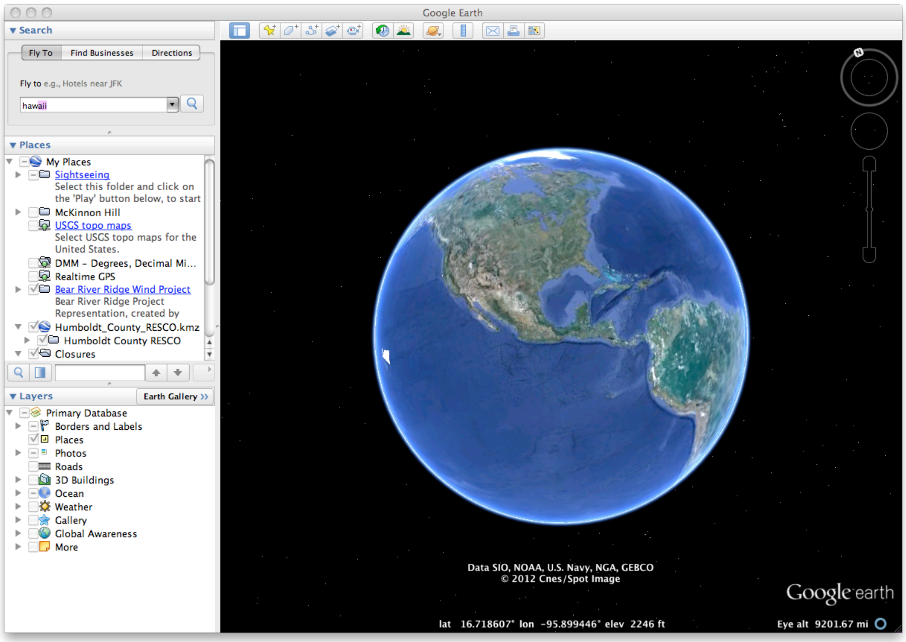
The Fishery Management Model Of Hawaii's Longline Fisheries (FMMHLF) allows the modeler to specify regions of the ocean for closure for specific fisheries over discrete time intervals each year. (The model uses the term "date closures" for this regulatory mechanism, because closure is triggered by the date instead of quotas.) The input file that provides the data to define date closures is a Google Earth KML ("Keyhole Markup Language") file. When the model runs, it looks for a file called "Closures.kml" in the experiment directory (the directory containing all of the input files for a model run).

This how-to section provides step-by-step instructions on how to create the "Closures.kml" file. Once this file is created and placed in the experiment directory, no other work is necessary to include the fishery closures in model runs. The file can later be edited to simulate alternative closures.

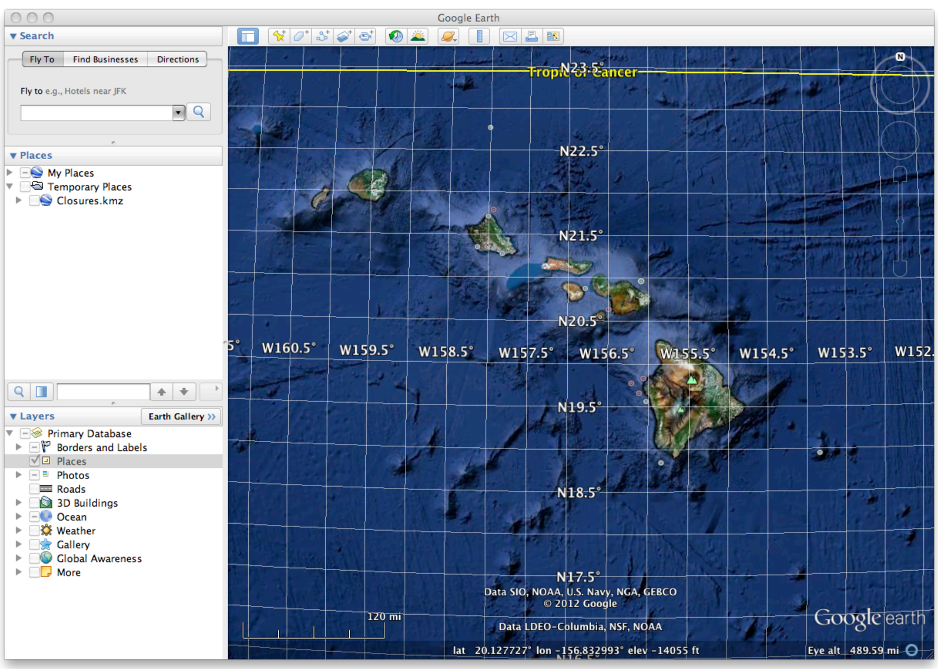
5.2 How to Create "Closures.kml" with Google Earth

You need Google Earth to do this; it is free software and can be downloaded at http://www.google.com/earth. There you can also find a number of tutorials, tips, and tricks for navigating the earth, creating animations, etc. The following instructions guide you through the process of creating polygons in Google Earth and exporting them in a format that can be read by FMMHLF. All recent versions of Google Earth should work; however, this tutorial was created with Google Earth 6.1, and details may be different for other versions (and may differ among Windows, MacIntosh, and Linux versions).

1. Open Google Earth and navigate to Hawaii (try typing "Hawaii" in the search box and hitting return).

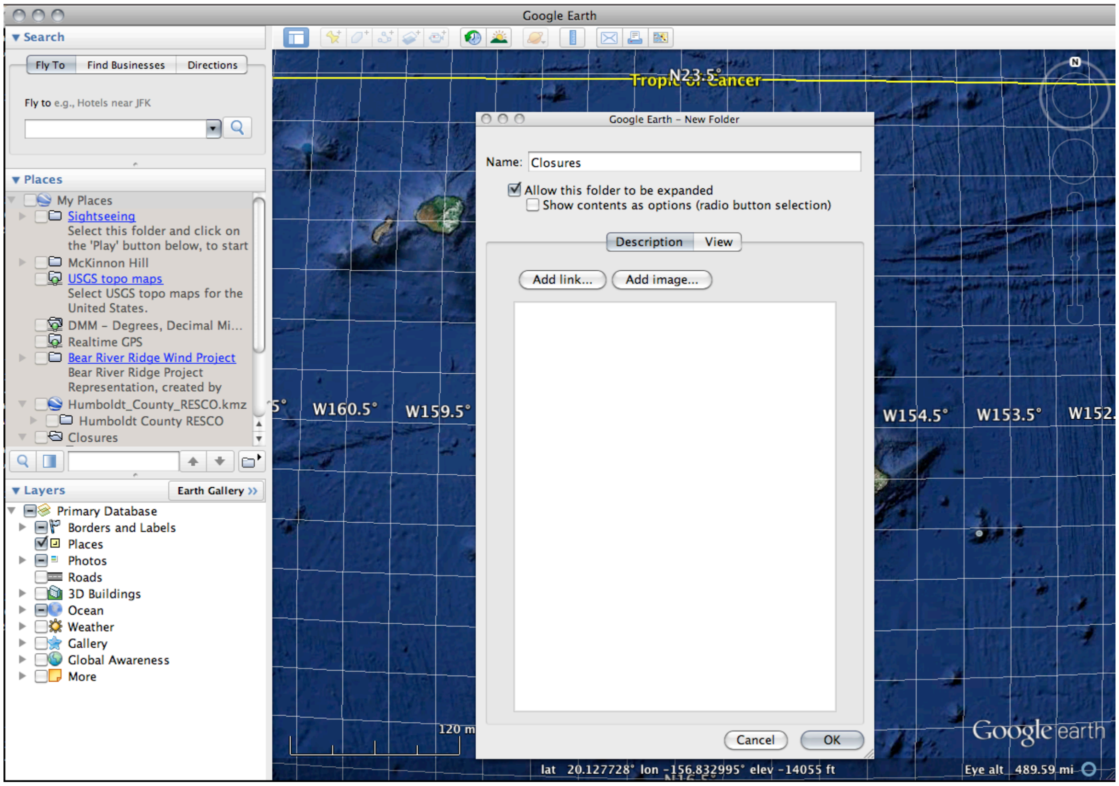


It can be helpful to see a latitude–longitude grid while you are creating the polygons. Click the "View" menu and then select "Grid".

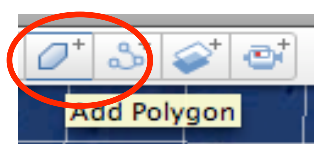


2. Regardless of whether you want to create one polygon or multiple polygons to specify closures, you need to create a folder in Google Earth to contain the polygons. In the "Places" pane on the left, right-click on either "My Places" or "Temporary Places" (temporary places are not saved when you quit, which is OK for this process but does not make it easy to go back and revise the fish closure polygons). Choose Add –> Folder. Name the folder

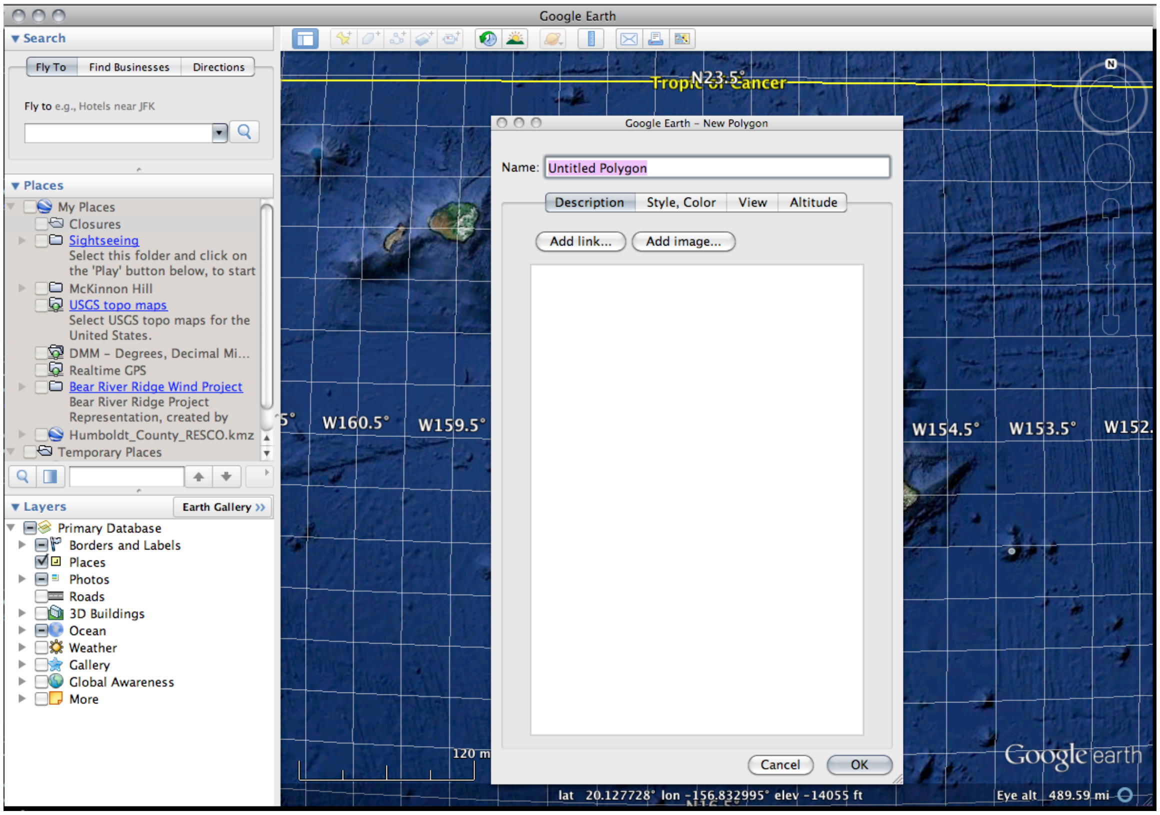
"Closures" or whatever you wish and click "OK". Make sure the folder appears in in the "Places" pane and is selected (highlighted).



3. Now click the polygon tool icon on the toolbar at the top of the window:



to open a new polygon dialog:

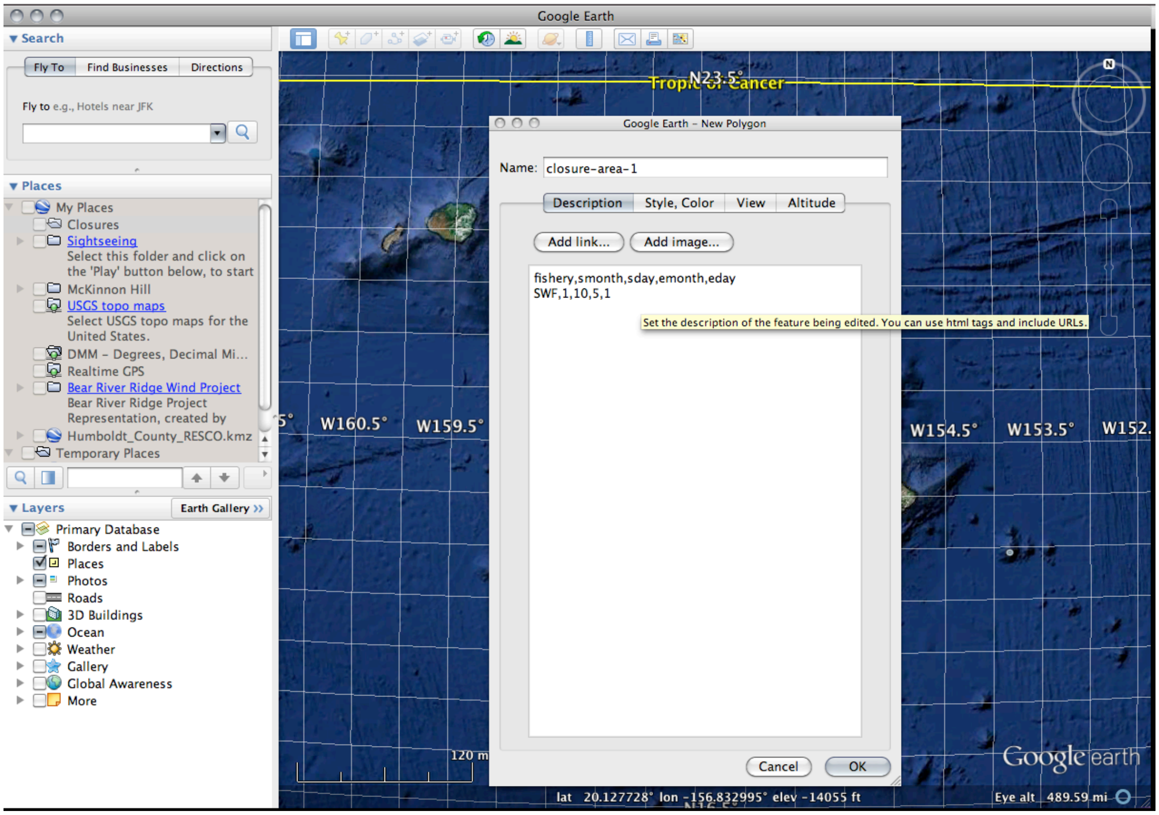


Name the polygon and, in the description field, enter the data specifying the fish species, start date, and end date of the closure. The data must be in a comma-separated format, with variable labels on the first line (exactly as in the following example) and values on the second line. One or more closures (combination of fishery and dates) can be specified per polygon. Valid values of the fishery variable can be found by starting a FMMHLF model run and clicking the button on its interfaced labeled “"Configure species and man areas"”. Currently valid fishery values are SWF for swordfish and TUNA for tuna fisheries.

An example of the description field (closing the swordfish fishery from January 1 to April 30, and both fisheries June 15 to July 31) is:

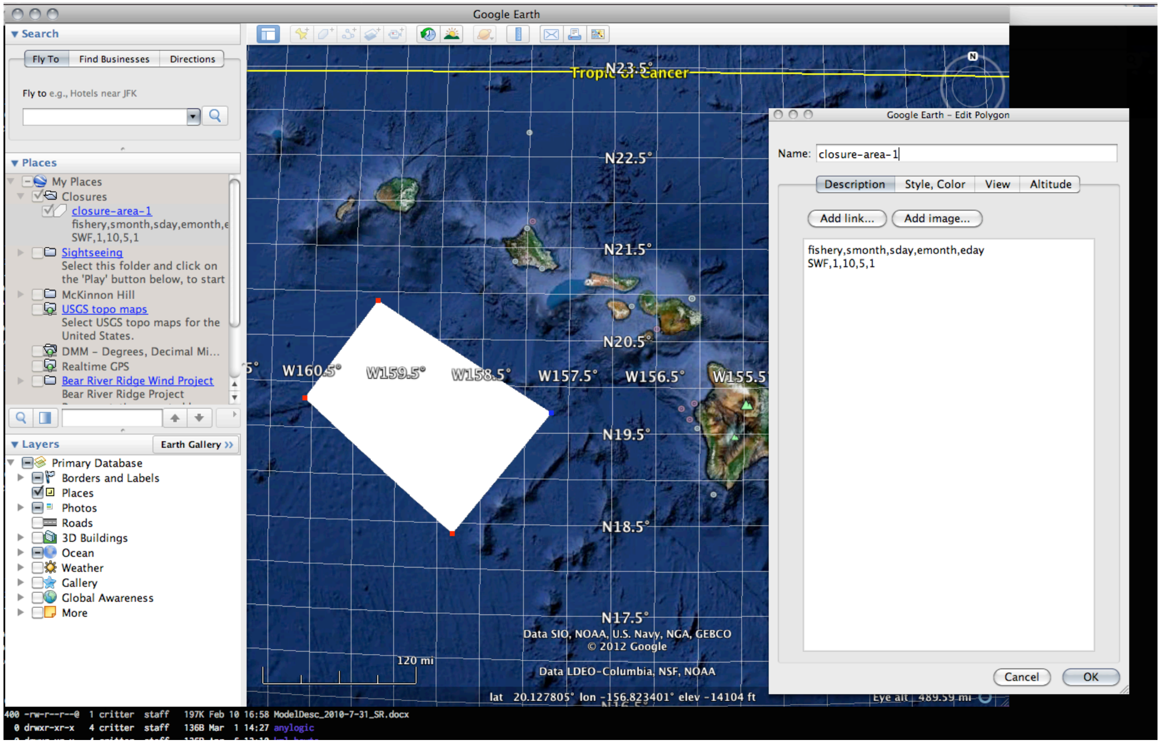
fishery,smonth,sday,emonth,eday
SWF,1,10,4,15
SWF,6,15,7,31
TUNA,6,15,7,31

Do not click “OK” yet, you still need to create the actual polygon.



4. Now draw the polygon by clicking on the map in the locations where you wish the vertices of the polygon to be. The vertices can be moved, added, or deleted later (see steps 5-6), so don't worry if you make a mistake.

   In drawing the polygon, keep in mind that the model delineates space as square grid cells 1° in width that are centered on the intersections of lines of latitude and longitude. A cell is included in a closure if its center is on or within the boundaries of the closure polygons we are now drawing. Hence, it is not important to draw the boundaries more precisely than 1°. Instead, the safest way to draw closure boundaries is to put them just outside the latitude-longitude intersection points defining the cells to be closed.

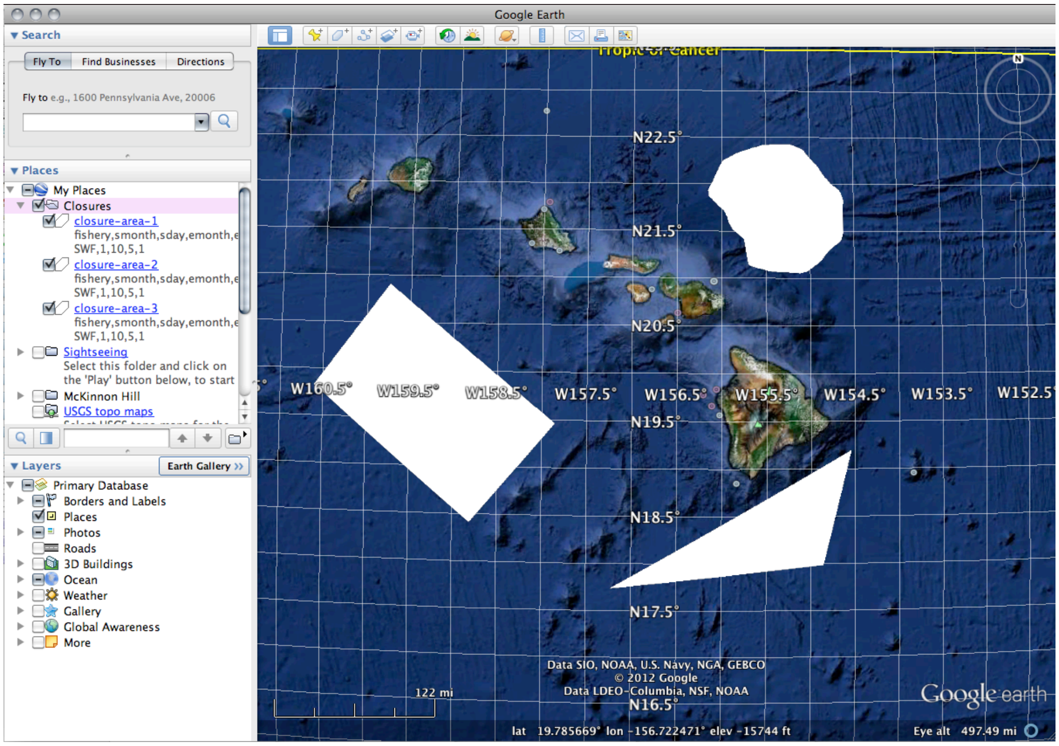


5. To move a vertex, hover the mouse over the point until the cross-hair turns into a finger, then click and drag.

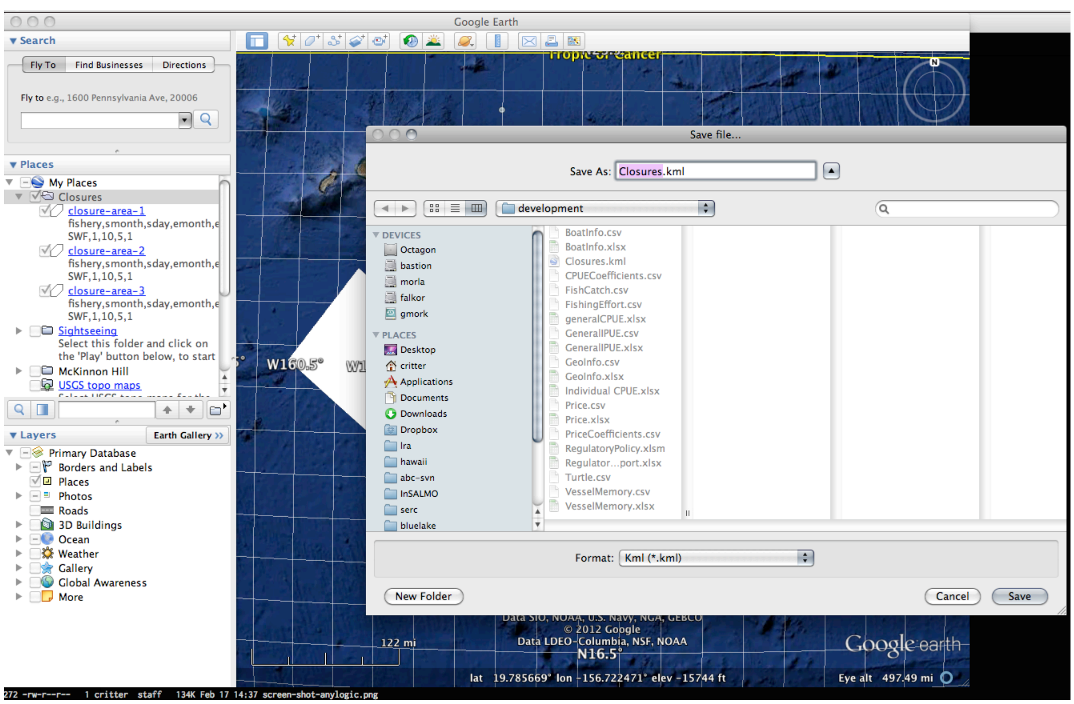
6. To add a vertex, first click on the existing vertex that is counter-clockwise to where you want the new one. Now click on the edge of the polygon to set a new point that you can drag to your desired location. To remove a point, first click on it, then right-click on it.

When the polygon is created, click OK in the polygon properties dialogue.

Repeat steps 4-7 for all of the closure areas you want in your experiment (there is no limit). Polygons can be any shape with any number of points defining the boundary.



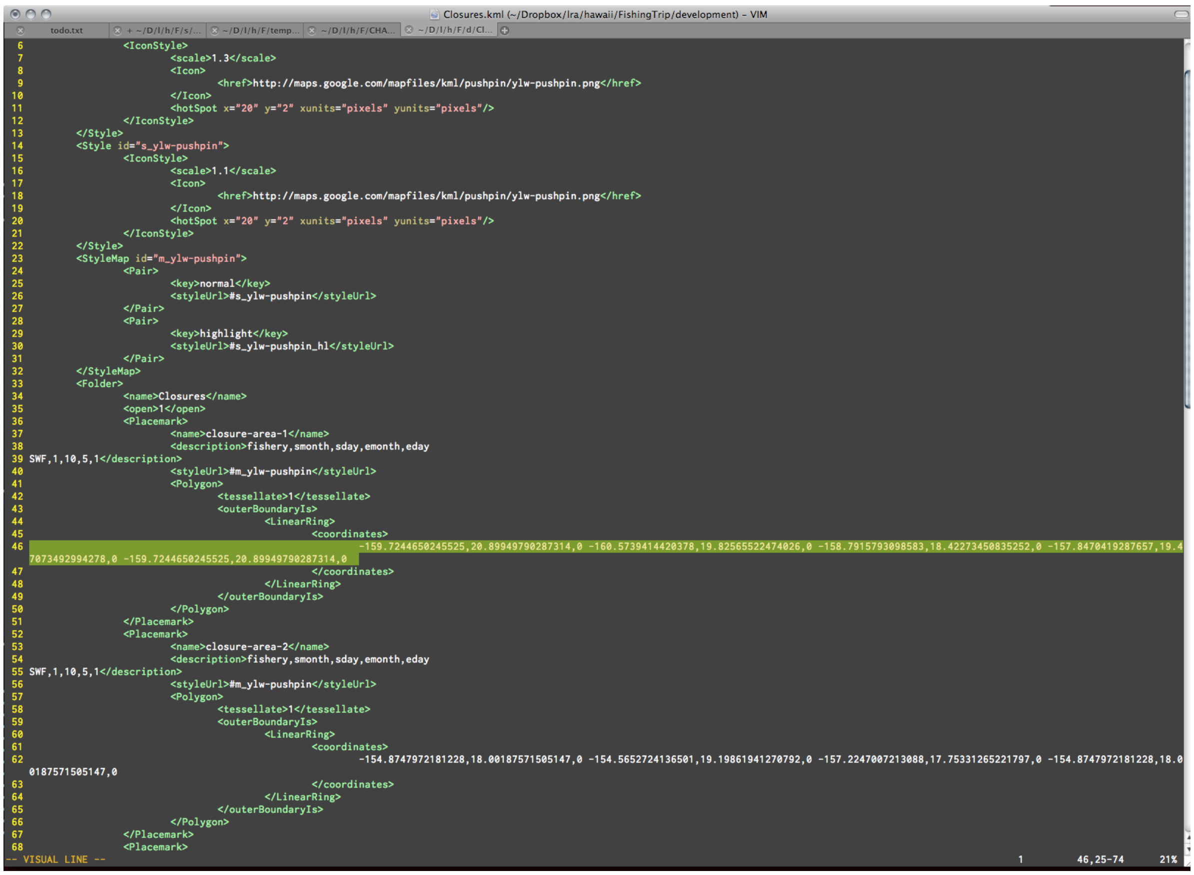
7. To export your closure areas as a KML file, right click on the folder containing the polygons and select "Save place as…" make sure that the format ".kml" is specified (not ".kmz"!) and that the file is named "Closures.kml" and placed in the directory containing your FMMHLF input files.



8. You're done! See Step 11 if you want to fine-tune the polygon vertices, and Step 12 to later revise or edit a closure polygon—either the shape of the polygon itself or the closure information (fishery, beginning and ending dates) associated with it.

5.3 How to Edit or Revise Closures.kml

1. You can open the KML file in a text editor (as illustrated in the following figure), very easily identify the polygons in the XML, and make manual edits to the latitudes and longitudes. A KML file is just a special kind of XML file, so it's ok to edit by hand and save as plain text.




```
            <IconStyle>
                <scale>1.3</scale>
                <Icon>
                    <href>http://maps.google.com/mapfiles/kml/pushpin/ylw-pushpin.png</href>
                </Icon>
                <hotSpot x="20" y="2" xunits="pixels" yunits="pixels"/>
            </IconStyle>
        </Style>
        <Style id="s_ylw-pushpin">
            <IconStyle>
                <scale>1.1</scale>
                <Icon>
                    <href>http://maps.google.com/mapfiles/kml/pushpin/ylw-pushpin.png</href>
                </Icon>
                <hotSpot x="20" y="2" xunits="pixels" yunits="pixels"/>
            </IconStyle>
        </Style>
        <StyleMap id="m_ylw-pushpin">
            <Pair>
                <key>normal</key>
                <styleUrl>#s_ylw-pushpin</styleUrl>
            </Pair>
            <Pair>
                <key>highlight</key>
                <styleUrl>#s_ylw-pushpin_hl</styleUrl>
            </Pair>
        </StyleMap>
        <Folder>
            <name>Closures</name>
            <open>1</open>
            <Placemark>
                <name>closure-area-1</name>
                <description>fishery,smonth,sday,emonth,eday
SWF,1,10,5,1</description>
                <styleUrl>#m_ylw-pushpin</styleUrl>
                <Polygon>
                    <tessellate>1</tessellate>
                    <outerBoundaryIs>
                        <LinearRing>
                            <coordinates>
                                -159.7244650245525,20.89949790287314,0 -160.5739414420378,19.82565522474026,0 -158.7915793098583,18.42273450835252,0 -157.8470419287657,19.4
7073492994278,0 -159.7244650245525,20.89949790287314,0
                            </coordinates>
                        </LinearRing>
                    </outerBoundaryIs>
                </Polygon>
            </Placemark>
            <Placemark>
                <name>closure-area-2</name>
                <description>fishery,smonth,sday,emonth,eday
SWF,1,10,5,1</description>
                <styleUrl>#m_ylw-pushpin</styleUrl>
                <Polygon>
                    <tessellate>1</tessellate>
                    <outerBoundaryIs>
                        <LinearRing>
                            <coordinates>
                                -154.8747972181228,18.00187571505147,0 -154.5652724136501,19.19861941270792,0 -157.2247007213088,17.75331265221797,0 -154.8747972181228,18.0
0187571505147,0
                            </coordinates>
                        </LinearRing>
                    </outerBoundaryIs>
                </Polygon>
            </Placemark>
            <Placemark>
```

2. One of the purposes of FMMHLF is to simulate the effects of alternative closure dates and areas. To change the closures, you can open an existing Closures.kml file and change either the polygon that defines the closed area or the dates and fisheries affected by the closure. On a computer where Google Earth is installed, simply open Closures.kml by (e.g.) double-clicking on it in Windows Explorer. This will open Google Earth and draw the closure polygons.

   You can then select and revise any closure polygon by right-clicking either on the polygon itself or on its name in the "Places" pane. Select "Properties" (or "Get info" on some operating systems) from the dialog that opens. This takes you to the polygon editing dialog where you can make changes exactly as in steps 3-7. When done, save the changes as in Step 9.

6 ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. SES-0918185. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Previous research projects that helped to lay the foundation for this work were funded by the Pelagic Fisheries Research Program (PFRP) under the Cooperative Agreement NA09OAR4320075 between the Joint Institute for Marine and Atmospheric Research (JIMAR) and the National Oceanic and Atmospheric Administration (NOAA). The views expressed herein are those of the authors and do not necessarily reflect the views of NOAA or any of its subdivisions. The authors thank Dr. Minling Pan of the Pacific Islands Fishery Science Center, NOAA Fisheries for her continuous support and advice during the development of this model. Finally, we thank Dr. Steven Gray of the Department of Natural Resources and Environmental Management, University of Hawaii at Manoa for his helpful comments and review.